Netherlands organization for applied scientific research

TNO-report

TNO Physics and Electronics Laboratory

P.O. B
2509
Oude
The H

Fax +31 70 328 09 61
Phone +31 70 326 42 21

report no. FEL-91-B201

copy no. 9

title

Calibrated Radiometer for Background Scanning - CARABAS

author(s):

Ing. R. van der Touw

date :

November 1991

classification	
title	: unclassified
abstract	: unclassified
report text	: unclassified
appendices A - E	: unclassified

no. of copies	: 15
no. of pages	: 86 (incl. appendices, excl.RDP & distr.list)
appendices	: 5

All information which is classified according to Dutch regulations shall be treated by the recipient in the same way as classified information of corresponding value in his own country. No part of this information will be disclosed to any party.

TNO

report no.	:	FEL-91-B201
title	:	Calibrated Radiometer for Background Scanning - CARABAS
author(s)	:	Ing.R. van der Touw
institute	:	TNO Physics and Electronics Laboratory
date	:	November 1991
NDRO no.	:	
no. in pow '91	:	22173
Research supervised by:		Dr.Ir. P.A.M. Jacobs
Research carried out by:		Ing.R. van der Touw

ABSTRACT (UNCLASSIFIED)

Thermal characteristics of targets and backgrounds determine IR detectability of the targets. Therefore not only knowledge of the thermal behaviour of targets is needed, but also of background elements like trees and grass. Long-term measurements are necessary to study time variations of apparent temperatures, covering all weather conditions. For this reason a dual wave-band scanner is developed for autonomous radiation measurements in the spectral regions 3-5 μm and 8-12 μm. This report describes system design and performance.

rapport no.	:	FEL-91-B201
titel	:	Calibrated Radiometer for Background Scanning - CARABAS
auteur(s)	:	Ing.R. van der Touw
instituut	:	*Fysisch en Elektronisch Laboratorium TNO*
datum	:	november 1991
hdo-opdr.no.	:	
no. in iwp '91	:	22173
Onderzoek uitgevoerd o.l.v.	:	Dr.ir.P.A.M. Jacobs
Onderzoek uitgevoerd door	:	Ing.R. van der Touw

SAMENVATTING (ONGERUBRICEERD)

Thermische eigenschappen van doelen en achtergronden bepalen de IR detecteerbaarheid van de doelen. Daarom is niet alleen kennis van het thermisch gedrag van doelen noodzakelijk, maar ook van achtergrond elementen zoals bomen en gras. Metingen over lange perioden zijn nodig om veranderingen in de tijd van schijnbare temperaturen te bestuderen, onder alle weersomstandigheden. Om die reden is een dual wave-band scanner ontwikkeld voor autonome stralingsmetingen in de spectrale gebieden 3-5 µm en 8-12 µm. Dit rapport beschrijft het systeem ontwerp en de technische mogelijkheden en prestaties.

TABLE OF CONTENTS

1 INTRODUCTION

Detection of targets is based on observation of contrasts with the background. In the infrared wave-bands (IR) these are temperature and emissivity contrasts. They are caused by differences in material and physical properties, like heat capacity and absorptivity. Military targets consist of non-natural materials. Very often there are heat producing systems inside, which cause extra warming up. This leads to differences in thermal behaviour and results in increased temperature contrasts. Consequently an increase of the detection probability follows. Reduction of the temperature contrast between object and background in principle can be achieved by adapting the surface temperature or using camouflage (influence on apparent temperature, chapter 2.). The materials must be chosen such that the contrast with the background will be minimized. Adapting of the correct type of camouflage requires knowledge of the behaviour of the background. The temperature behaviour depends strongly on the weather. To study this behaviour, a scanning radiometer was developed, to perform autonomous measurements of apparent temperatures according to a preset time interval. The name of this scanner is CARABAS, an acronym for CAlibrated RAdiometer for BAckground Scanning. Together with the registered meteorological parameters (chapter 6), the apparent temperatures form a valuable database to study time- and weather dependent temperature behaviour of background elements.

2 APPARENT TEMPERATURE

The principle of IR detection is based on observation of radiance differences. In this chapter the theory concerning the apparent temperature will be explained.

The spectral emission $W(\lambda,T)$ (monochromatic emissive power) of an object with absolute temperature T is given by Planck's formula, (1).

$$W(\lambda,T) = \frac{c_1}{\lambda^5 \{\exp(c_2/\lambda T)-1\}} \qquad (W\cdot m^{-2}\cdot\mu m^{-1}) \qquad (1)$$

with $c_1 = 3.74\text{x}10^8$ $W\cdot m^{-2}\cdot\mu m^{-4}$
and $c_2 = 1.439\text{x}10^{-2}$ $m\cdot K$ (λ in μm).
The radiation Q in the spectral waveband λ_0 is

$$Q = \int_{\lambda_0} W(\lambda,T)\, d\lambda \qquad (W\cdot m^{-2}) \qquad (2)$$

The radiation of an object, Q_{obj}, and the environmental radiation, Q_{env}, are determined by equations 1 and 2. The total radiation of an object, Q_{tot}, consists of an emissive part, $\varepsilon.Q_{obj}$, and a reflective part radiation, $\rho.Q_{env}$, i.e.

$$Q_{tot} = \varepsilon.Q_{obj} + \rho.Q_{env} \qquad (W\cdot m^{-2}) \qquad (3)$$

ε is the emissivity and ρ the reflectivity. The emissivity indicates a surface capacity to emit radiation and depends on the surface properties ($0 \le \varepsilon \le 1$). If α is the absorptivity and τ the transmissivity, then

$$\alpha + \rho + \tau = 1 \qquad (-) \qquad (4)$$

There is no transmission through solid materials and according to Kirchhoff's law the emissivity is equal to the absorptivity at wavelength λ, so eq. 4 becomes:

$$\rho = 1 - \varepsilon \qquad (-) \qquad (5)$$

In case of a black body $\varepsilon = 1$. The total emittance and its spectral distribution only depends on the temperature of the black body.

From the preceding it follows that detected radiation Q_{tot} can be expressed in terms of a temperature that may differ from the actual surface temperature due to different emissivities. This temperature is the apparent temperature T_{app}. In the spectral band λ_0, T_{app} is solved from equation 6, using eq. 2 and 3:

$$\int_{\lambda_0} W(\lambda,T_{app})\, d\lambda = \int_{\lambda_0} \tau_\alpha \{\varepsilon\, W(\lambda,T_{obj}) + \rho\, W(\lambda,T_{env})\}\, d\lambda \qquad (W\cdot m^{-2}) \qquad (6)$$

In this equation T_{obj} is the surface temperature of the object and τ_a the transmissivity of the atmosphere. $W(\lambda,T_{env})$ is the contribution of environmental radiation, predominantly sky- and scattered atmospheric radiation. T_{env} is an average "environmental" temperature. This is for example cloud base or sky temperature.

Q_{tot}, and consequently T_{app}, is determined by the temperature and emissivity of the surface (eq. 1 and 3). From eq. 6 it follows that it is possible to control Q_{tot} by changing the emissivity and consequently increasing reflection, mostly of cold sky radiation.

Not all parts of the IR spectrum are useful for IR measurements because of atmospheric attenuation. Due to molecules like H_2O and CO_2, IR radiation will be absorbed in the atmosphere, for example in the 5-8 μm region predominantly by H_2O. The regions where the atmosphere is transparent (windows) are 3-5 μm and 8-12 μm.

Differentiating the Planck equation with respect to λ and setting the result equal to zero, Wien's displacement law is obtained:

$$\lambda_{max}.T = 2893 \qquad (\mu m\cdot K) \qquad (7)$$

where λ_{max} is the wavelength at maximum emission, expressed in micrometers . A plot of this relation is shown in figure 2.1. Within a natural scene most objects, e.g. vegetation, will have temperatures close to that of the ambient air temperature, say 290 K, which will result in λ_{max} =

10 μm, which occurs in the centre of the 8-12 μm window. Therefore the atmospheric window 8-12 μm is most suitable to measure radiation of natural sources.

The material properties of an object, like density, heat capacity and emissivity, determine the physical temperature of the surface. Evaporation at the surface will influence the temperature as well.

3 THE CARABAS SYSTEM

The CARABAS is a calibrated dual waveband radiometer, suitable for long-term outdoor operation. In this chapter optics, scanning mechanism, internal calibration and system performance will be described.

3.1 Optics and Detectors

To measure radiation of a single object only, instead of an average value over different objects, a small instantaneous field of view (IFOV) is necessary. On the other hand a reasonable area must be covered when measuring over short distances. Therefore the IFOV of the CARABAS is 1°x1° using Cassegrain optics, a combination of an hyperbolic and parabolic mirror. Figure 3.1 shows these Cassegrain optics.

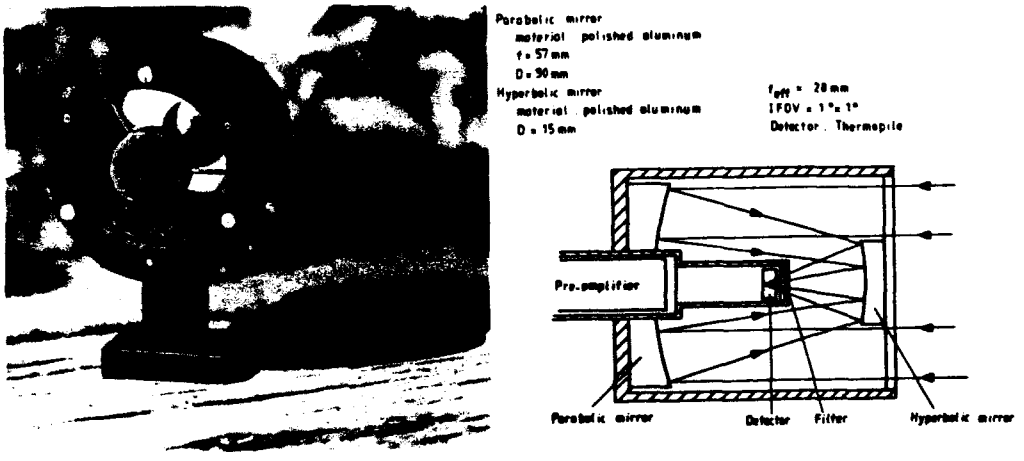

Fig. 3.1 The Cassegrain optics in the CARABAS.

The equations of the hyperbolic and parabolic mirrors are

$$\frac{x^2}{49} - \frac{y^2}{371.25} = 1 \qquad (8)$$

respectively

$$x = \frac{y^2}{228} \qquad (9)$$

The effective focal distance of this combination is 28 mm, projecting a parralel beam with a diameter of 30 mm on the detector.
The mirrors are made of polished aluminium.

Two thermopile detectors (Laser-Optronic, type 5M) are used, since thermopiles do not need cooling for optimum performance. The responsivity of the thermopile detectors is wavelength independent. By using interference filters the received radiation is limited to the windows 3-5 μm respectively 8-12 μm.
The filter characteristics are shown in figure 3.2, where transmission is plotted versus wavelength.

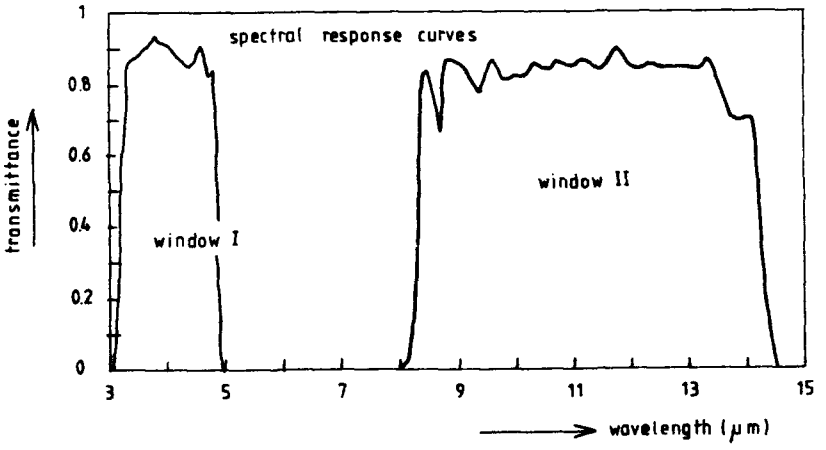

Fig. 3.2 Transmission versus wavelength of the interference filters used in the CARABAS.

3.2 Calibration procedure and scanning mechanism

one of the most important features of the CARABAS is the internal calibration, before the actual measurement. This reduces the influence of electronic drift on the results. Therefore not only the radiation of the environmental object is measured, but also of three internal calibration sources. These blackbody sources have temperatures of 0, +3 en +9 °C relative to the scanner housing temperature. They are mounted close to the detectors, at the inside of the scanner housing see fig. 3.3. The temperature of the scanner housing will be close to ambient air temperature.
The temperatures of the sources are read by Pt-100 resistors and continuously adjusted. By fitting a least square curve, using the values of the three calibration sources (volts, $V_c(i)$, versus temperatures, $T_c(i)$, i=1,2,3), the apparent temperature of the unknown element T_{app}^x can be obtained, using:

$$Q_c(i) = \int_{\lambda_0} T_\lambda \, W[\lambda, T_c(i)] \, d\lambda = a \, V_c(i) + b \qquad (W.m^{-2}) \qquad (10)$$

where $Q_c(i)$ is the calibrated radiation for each blackbody and T_λ the transmission of the filters. T_{app}^x can be solved from

$$a \, V_x + b = \int_{\lambda_0} T_\lambda \, W(\lambda, T_{app}^x) \, d\lambda \qquad (W.m^{-2}) \qquad (11)$$

where V_x is the measured detector voltage.
CARABAS measurements are carried out over short ranges (up to 100 meters) and therefore the transmissivity τ of the atmosphere is assumed to be 1. Per position the sequence of measurements of both detectors, as shown in fig. 3.3, are identical.

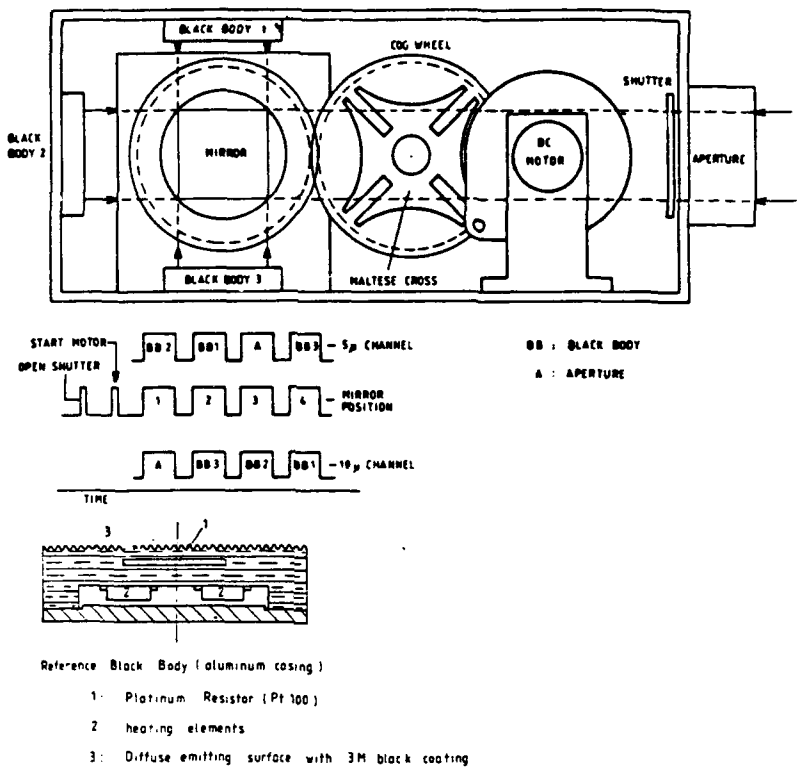

Fig. 3.3 Sequence of measurements with Maltese cross of the CARABAS.

By means of a double sided mirror both detectors make simultaneous measurements. This mirror is mounted in a rotating drum, at a 45° angle to the optical axis. Four discrete steps of 90° rotation around the drum axis are performed per complete cycle (see figure 3.3 and 3.4.). The mirror remains in each position for 1 second. To increase accuracy the detector signal is integrated over 0.75 seconds. In this way both detectors measure the environmental object and the three calibration sources within 4 seconds, aligned optically to measure exactly the same areas.
To obtain four exact discrete steps a Maltese cross is used (see fig. 3.3), driven by a DC motor.
Photo 3.5 shows the CARABAS interior.

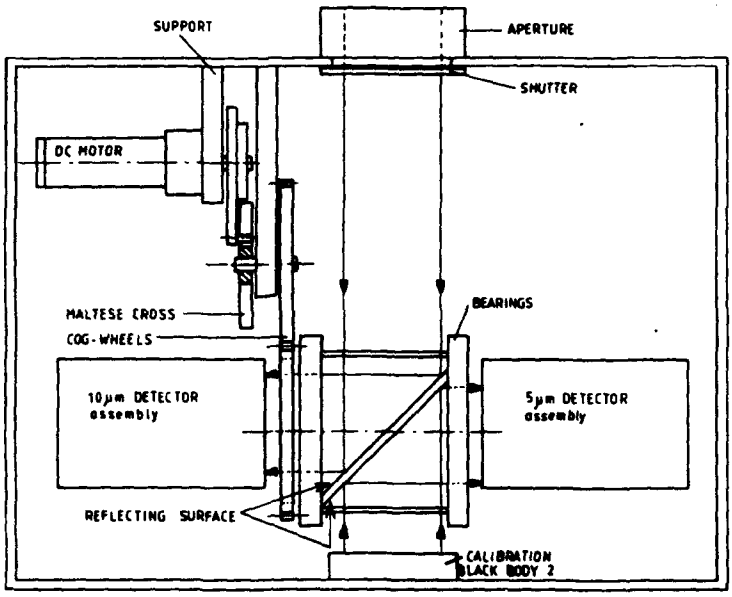

Fig. 3.4 Optical path in the CARABAS.

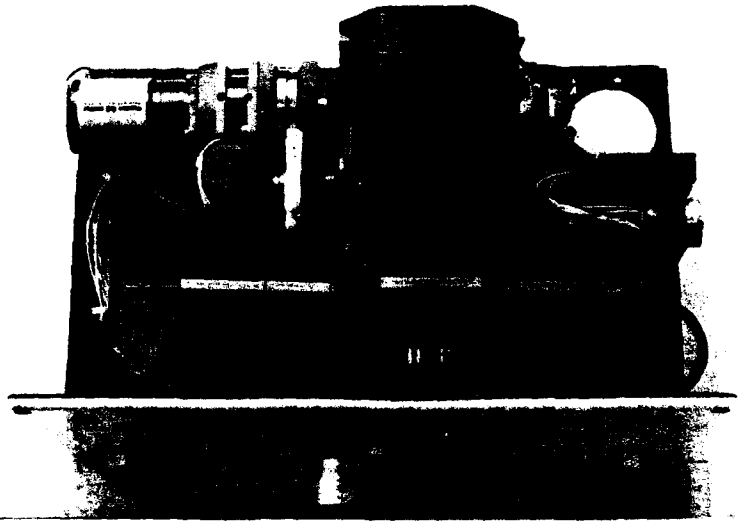

Photo 3.5 The CARABAS interior.

3.3 Scanner housing

To avoid rain, dirt or small insects from entering the CARABAS, a shutter closes automatically the entrance pupil when not operating. A tube is placed in front of the entrance to avoid rain coming in during measurements. Protection against solar heating is achieved by mounting reflecting shields at all sides of the scanner housing. All outer parts of the scanner are painted white. On top of the sensor housing a CCD camera can be mounted to observe the positions, for example to record or define locations.

3.4 Control system

The CARABAS system is scanning autonomous according to a preset timetable and position sequence. All movements are remotely controlled by a portable IBM Personal Computer. This PC is provided with a 12 bits, 100 kHz LABMASTER data acquisition system (Scientific Solutions Inc.) and a 30 Mb storage capacity. The measured values of radiation are converted to apparent temperatures (degrees Celcius) and stored together with the time of measurement. One day with 30 positions and a scan interval of 15 minutes results in a file of 90 kb.
The LABMASTER is connected to a separate control unit. From this unit the sensor can be directed by joy-stick as well.
The (single) cable between the control unit and the CARABAS is 10 meter long. The control unit and the PC are placed in an air-conditioned cabin.
In case of a power failure a no-break power back-up system (Inmelec MS 600) is activated. It provides the CARABAS system for about 10 minutes with power in order to stop measurements correctly, return scanner head to first position and close files and to restart automatically after power has returned.

3.5 System performance

The Noise Equivalent Temperature Difference (NETD) of the CARABAS is < 0.04 °C at 20 °C for both channels. The temperature accuracy in 3-5 μm is < 0.2 °C and in 8-12 μm < 0.1 °C.

The CARABAS can be rotated over 340° azimuth and -90° to +90° in elevation. The maximum accuracy of positioning is ± 0.09 ° in both elevation and azimuth. For practical reasons the step size of the sensor head can be set to a multiple of 0.09 ° to reduce operation time. The CARABAS is presently set to ± 0.9 ° (step size 10), scanning 30 positions in about 5 minutes,

covering the whole range of angles with optimized scan sequence (see chapter 4.1; highest optimization priority is azimuth, followed by elevation).
The scanner can operate outdoors for over a year without service. Photo 3.6 shows the CARABAS on a measurement site.

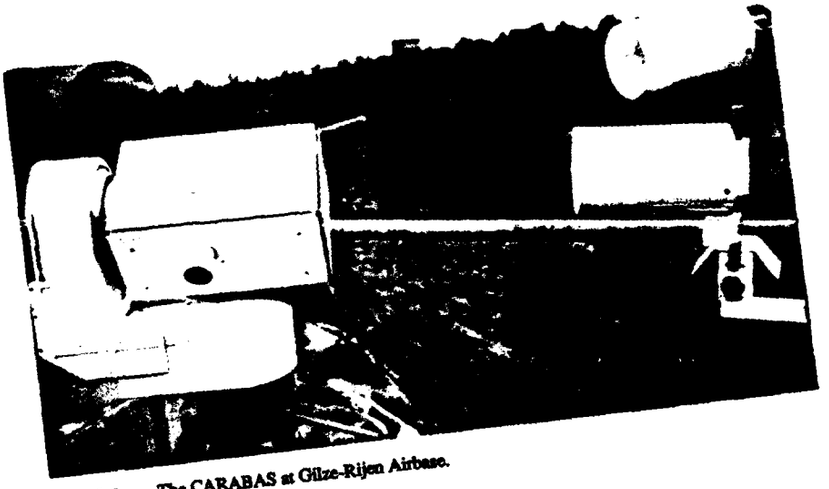

Photo 3.6 The CARABAS at Gilze-Rijen Airbase.

4 CARABAS CONTROL

Three programmes are developed in BASIC language to operate the CARABAS system. The program "DEFPOINT" is used to define measurement positions, "CARABAS" performs the actual measurements and data storage and "EVAL" gives a first analysis of measurements to check the CARABAS operation.

4.1 Define positions

The programming of the measurement positions is done with a program called "DEFPOINT". Using a joy-stick or keyboard, the CARABAS can be positioned. The X and Y coordinates (azimuth and elevation) will be fixed and read by push button control. A CCD camera can be mounted on top of the CARABAS scanner to record the selected positions. When all measurement points are defined, the scanning sequence can be optimized to minimize the measurement cycle time. The sequence of positions is stored in a file. This position file can be adjusted easily by inserting or deleting measurement locations.
A position file name has the format "mmmddyyP" (like JAN0191P, meaning the creation date, January 01, 1991, with addition P for Position file) and contains the following information:

number of positions	(p)
date	(mm-dd-yy)
position information (p blocks)	
object description	(30 characters)
status specification	(30 characters)
x-coordinate	-
y-coordinate	-
elevation angle	(degrees)
azimuth angle	(degrees)
height	(cm)
distance	(m)
emissivity	(-)
background description	(30 characters)
status specification	(30 characters)
position code	(11 characters)
end block (p)	

The program generates a position code for each position, giving complete and uniform descriptions of the background elements. These codes are generated from user input. They contain information about object description, elevation, azimuth, physical condition and appearance.
A position code consists of 11 digits. If "aabcdefghhi" represents the code, then

aa = object description
b = status specification of the object
c = elevation angle
d = azimuth angle
e = height
f = distance
g = emissivity
hh = background description
i = status specification of the background

Table 4.1 Descriptions and specifications of objects.

01	AGRICULTURE	1	GREEN
02	GRASS	2	FALLING
03	BUSHES-CONIFEROUS	3	BARE
04	BUSHES-DECIDUOUS	4	BUDS
05	TREES-CONIFFEROUS	5	GROWING-NORMAL
07	TREES-DECIDUOUS	6	FRESH-CUT
"		7	CRUSHED
11	CONCRETE	8	COVERED WITH MUD
12	ASPHALT	9	COVERED WITH SNOW
13	GRAVEL	0	NO SUB-TITLE
14	SAND-BARE		
15	SOIL-BARE		
16	MUD		
"			
21	SLOPE-UP-HILL		
22	SLOPE-DOWN-HILL		
23	NATURAL-WASTE		
"			
31	CAMOUFLAGE-NET		
32	CAMOUFLAGE-SCREEN		
33	CAMOUFLAGE-ADAPTIVE		
"			
41	WATER		
42	SNOW		
43	MUDDY-SNOW		
44	SKY		
45	HORIZON		
"			
91	BLACKBODY + 3 DEG		
92	BLACKBODY + 13 DEG		

The emissivity is measured with an infrared reflectometer (ref 1). Table 4.1 shows a list of existing descriptions and status specifications for objects and their background. The

corresponding numbers can be found in the position code (digits aa and hh). The status specification gives additional information, most temporary, like covered by snow or mud (digits b and i). The X- and Y coordinates of the scanner are only of importance during the measurements and will therefore not be included in the position code. The orientation of objects is defined by their azimuth and elevation angles, as shown in figure 4.1.
Appendix A shows the listing of the programme "DEFPOINT" and in Appendix B an example is given of a position file.

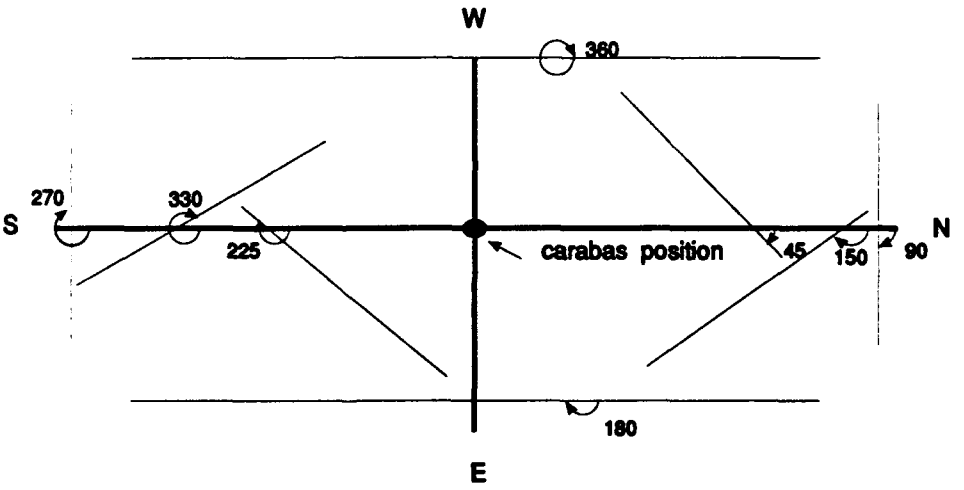

Fig. 4.1 Definition of object orientations in position file.

The position file is updated once a week if any changes have occurred.
To check the positioning accuracy a testrun can be performed, using the CCD camera, in which the consecutive positions are displayed for approval. If necessary their descriptions can be adjusted.

4.2 System operation

The software controls the positioning and measuring sequence, and also handles error recovery. The program to do this is called "CARABAS".
The program is initialized, defining the following parameters:

1. Start date
2. Stop date
3. Start time
4. Stop time
5. Measurement sequence time interval
6. Name of position file
7. Name of data file

Before starting the measurement cycle, an initialization file is created, containing these parameters, in case an automatic restart after power failure is required. A listing of the program "CARABAS" is given at Appendix C.

4.2.1 Data collection

The program "CARABAS" performs the complete measurement cycle. It starts and stops measurements according to the preset timing parameters. The position information is read from the position file and used to direct the sensor to each location. The coordinates of positions are expressed in units of 0.09°. The program feedback recognizes when the sensor is in correct position. The actual measurements are executed by control of the position of the internal mirror. After scanning all positions, the sensor pupil is closed by a shutter and the sensor returns to the first position. At this time data is available as detector voltages and Pt-100 voltages per channel and per position.

Using equations 10 and 11, the curve fit parameters a and b are determined and the unknown apparent temperature is calculated. This is repeated for each position. The calculated temperatures are stored on disk. The CARABAS file name has the format "mmmddyyC" (like JAN0191C, meaning the creation date, January 01, 1991, with addition C for CARABAS file) and every day a new file with a corresponding name is created.

Data is stored per scan interval (15 minutes default) in the following format:

position file name	(JAN0191P)
number of positions	(p)
date	(mm-dd-yy)
carabas position data (p blocks)	
time	(hh:mm:ss)
T_{app} in 3- 5 μm band	(°C)
T_{app} in 8-12 μm band	(°C)
end block (p)	

The precise observation time is of significance as CARABAS data will be correlated to weather data. Appendix D shows an example of a set of results.

4.2.2 Error recovery and control

Internal or external errors might occur during running of the program. To minimise the loss of data as a result of this errors, the program checks for power failure, I/O- and hardware (sensor) errors. Also the sensor positioning is checked continuously during a measurement cycle. Detection and if possible recovery makes normal execution of the program possible after an error has occurred.

Two external blackbodies with a constant temperature difference are used to control operation. This is done by the user. As an option this control can be inserted in the program to check the quality of the measurements automatically.

Temperatures of both spectral bands of up to four positions and up to 20 previous scans can be displayed during running of the program.

4.3 Evaluation

The first analysis of measured apparent temperatures can be performed by a program called "EVAL". The program reads CARABAS data. Average and standard deviation of selected objects are calculated and temperatures versus time or frequency of occurrence are printed or plotted. This program is meant for quick look. A second program, called "CLEANCAR", is available to edit CARABAS data in case of individual data distortions.

5 MEASUREMENT FACILITIES

At the airbase Gilze-Rijen in the Netherlands a suitable test site for background measurements was found (photo 5.1). The CARABAS is mounted on a platform at 7 meters above ground level, placed beside an air-conditioned cabin with the control units. Besides environmental elements, several camouflage materials are measured as well. The location at the airbase is wide, quiet and closed to public. It is therefore suitable to test new developed materials as well.

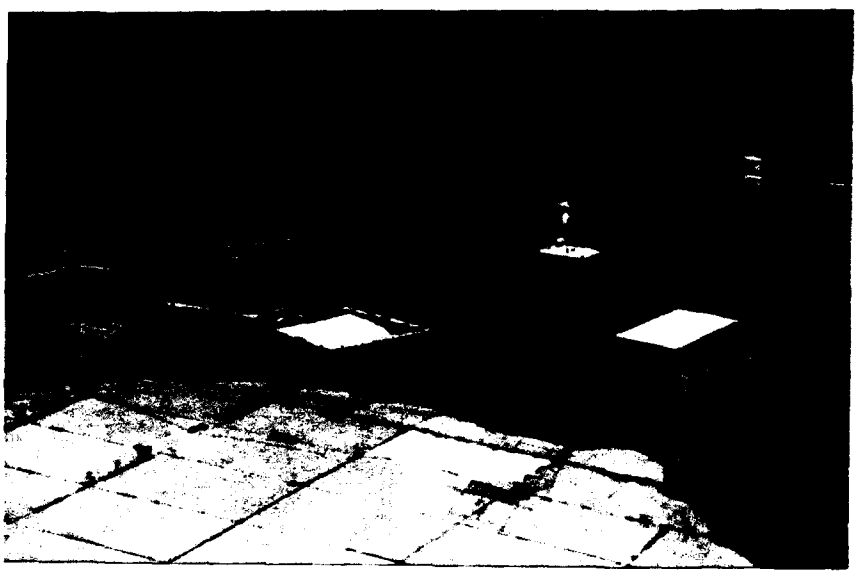

Photo 5.1 Test site at Airbase Gilze-Rijen.

The CARABAS is placed in the field in November 1989. For over a year it has remained there without service. Only once a small bird entered the sensor head and was chopped by the rotating drum.

To check if there are any changes of the measured background elements, the site is visited on a weekly basis, also for checking system performance and general maintenance i.e. cleaning of the optical parts. During these visits the position file is updated to match the actual situation. The current set-up will be maintained until April 1991. By then it is expected that all meteorological conditions are covered.

6 METEOROLOGICAL DATA

A weather station, located close to the CARABAS, collects the 5 minutes average of eight meteorological observations. The quantities measured are: air temperature, relative humidity, air pressure (all 1.5 m above ground level), global irradiance (0.3 - 3 μm.), "global" IR irradiance (pyrgeometer, 3 - 50 μm.), wind speed and direction (at 10 m above ground level) and precipitation (ref 2). The meteorological data is stored in a separate file, named "mmmddyyM" (like JAN0191M, meaning the creation date, January 01, 1991, with addition M for Meteo file), per period of 5 minutes. This file contains the following information:

date	(mm-dd-yy)
time	(hh:mm:ss)
number of scans in last period	(n)
meteo data block (8 parameters)	
meteo parameter i	(...)
number of valid scans of parameter i	(n_i)
end block	(8)

An example of meteo results is given in Appendix E.
A description of the weather station can be found in report FEL-91-B202, [ref. 2].

7 CONCLUSIONS

The CARABAS system can give an impression of the temporal IR behaviour of objects in two spectral bands, 3-5 μm and 8-12 μm. The measurements of apparent temperatures of any object can be carried out autonomously over a period of at least a year with the CARABAS system. The objects can be backgrounds as well as targets. Also camouflage efficiency can be analysed. There is data available now of about 24 different objects, from the period april 1990 to april 1991. The temperature accuracy is better then 0.2 °C and the maximum position accuracy 0.09°, unchanged after two years of performance in the field. With the IFOV of 1° the size of measured object surfaces can be varied by changing the distance.

8 RECOMMENDATIONS

One of the parameters of importance is the solar absorptivity. No equipment is available yet to measure this parameter easily. A pyrheliometer combined with measurements will give useful suplementary information. Extension with a visual camera on top of the CARABAS scanner offers the possibility to combine IR data with visual data. The data acquisition (of CARABAS as well as meteorological data) should be limited to only one PC and offering the possibility of on-line processing.

The CARABAS system is also useful in industrial environments to perform remotely controlled temperature measurements in hazerdous areas or complex installations that cannot be reached easily.

9 REFERENCES

[1] Description of an infrared reflectometer,
Ir. A.N. de Jong

[2] An Automatic Meteorological Station,
Ing. R. van der Touw, FEL-91-B202,
October 1991, TNO-FEL, The Hague

A.N. de Jong
(groupleader)

Ing. R. van der Touw
(author)

DEFPOINT software listing

```
*   '                    PROGRAMME NAME  :  D E F P O I N T . B A S

    '                    THIS PROGRAMME DEFINES THE COORDINATES OF THE
    '                    MEASUREMENT POINTS, AS THEY ARE SCANNED BY THE
    '                    RADIOMETER "CARABAS".
    '                    THE TABLE, CONTAINING THESE COORDINATES IS WRITTEN
    '                    TO AN INTERACTIVELY GIVEN FILE AND STORED ON DISK.

    '                    PARAMETERS USED IN THIS PROGRAMME ARE:

    'X(J),Y(J)   :       COORDINATES OF MEASUREMENT POINT J
    'X0  ,Y0     :       COORDINATES OF THE ORIGIN

    '                    A/D CONVERTER CHANNELS "DAC":
    'DAC = 0     :       AZIMUTH MOTOR   I.E. X-COORDINATE
    'DAC = 1     :       ELEVATION MOTOR   I.E. Y-COORDINATE

    const MAXDIM = 30
    dim X (MAXDIM) ,Y (MAXDIM) ,POST$ (MAXDIM),ELPOST$ (MAXDIM)
    dim HX (MAXDIM),HY (MAXDIM),H$ (MAXDIM)    ,HL$ (MAXDIM)
    dim TETA (MAXDIM),AZIM (MAXDIM),H (MAXDIM),RAN (MAXDIM),EPS (MAXDIM)
    dim TETAC(MAXDIM),AZIMC(MAXDIM),HC(MAXDIM),RANC(MAXDIM),EPSC(MAXDIM)
    dim SCREEN$ (MAXDIM),ELSCR$ (MAXDIM),POSCODE$ (MAXDIM)
    dim HS$ (MAXDIM)    ,HE$ (MAXDIM)    ,HP$ (MAXDIM)
    dim INFO$ (MAXDIM)
    dim MON$(12)
    MON$ (1) = "JAN"
    MON$ (2) = "FEB"
    MON$ (3) = "MAR"
    MON$ (4) = "APR"
    MON$ (5) = "MAY"
    MON$ (6) = "JUN"
    MON$ (7) = "JUL"
    MON$ (8) = "AUG"
    MON$ (9) = "SEP"
    MON$ (10)= "OCT"
    MON$ (11)= "NOV"
    MON$ (12)= "DEC"
    def fn DAG$(DAT$) = MID$(DAT$,4,3)+MON$(VAL(LEFT$(DAT$,2)))+RIGHT$(DAT$,5)
    def FNNUM$(X)
      X$ = STR$ (X)
      LENGTH = LEN (X$)
      if LEFT$ (X$,1) = "-" or LEFT$ (X$,1) = " " then
        LENGTH = LENGTH - 1
      end if
      FNNUM$ = RIGHT$ (X$,LENGTH)
    end def
    dim CODE$(30)
    CODE$ (1)  = "01    AGRICULTURE         "
    CODE$ (2)  = "02    GRASS               "
    CODE$ (3)  = "03    BUSHES-CONIFEROUS   "
    CODE$ (4)  = "04    BUSHES-DECIDUOUS    "
    CODE$ (5)  = "05    TREES-CONIFEROUS    "
    CODE$ (6)  = "06    TREES-DECIDUOUS     "
    CODE$ (7)  = ""                             '0*    TYPE:VEGETATION
    CODE$ (8)  = "11    CONCRETE            "
    CODE$ (9)  = "12    ASPHALT             "
    CODE$ (10) = "13    GRAVEL              "
    CODE$ (11) = "14    SAND-BARE           "
    CODE$ (12) = "15    SOIL-BARE           "
    CODE$ (13) = "16    MUD                 "
    CODE$ (14) = ""                             '1*    TYPE:UNSTRUCTURED
    CODE$ (15) = "21    SLOPE-UP-HILL       "
    CODE$ (16) = "22    SLOPE-DOWN-HILL     "
```

```
CODE$ (17) = "23    NATURAL-WASTE       "
CODE$ (18) = ""                                  '2*    TYPE:STRUCTURED
CODE$ (19) = "31    CAMOUFLAGE-NET      "
CODE$ (20) = "32    CAMOUFLAGE-SCREEN   "
CODE$ (21) = "33    CAMOUFLAGE-ADAPTIVE"
CODE$ (22) = ""                                  '3*    TYPE:CAMOUFLAGE
CODE$ (23) = "41    WATER               "
CODE$ (24) = "42    SNOW                "
CODE$ (25) = "43    MUDDY-SNOW          "
CODE$ (26) = "44    SKY                 "
CODE$ (27) = "45    HORIZON             "
CODE$ (28) = ""                                  '4*    TYPE:NATUR-ELEMENTS
CODE$ (29) = "91    BLACKBODY   + 3 DEG"
CODE$ (30) = "92    BLACKBODY   +13 DEG"
dim ELCODE$(10)
ELCODE$ (1) = "1  GREEN                 "
ELCODE$ (2) = "2  FALLING               "
ELCODE$ (3)   "3  BARE                  "
ELCODE$ (4) = "4  BUDS                  "
ELCODE$ (5) = "5  GROWING-NORMAL        "
ELCODE$ (6) = "6  FRESH-CUT             "
ELCODE$ (7) = "7  CRUSHED               "
ELCODE$ (8) = "8  COVERED WITH MUD      "
ELCODE$ (9) = "9  COVERED WITH SNOW     "
ELCODE$ (10)= "0  NO SUB-TITLE          "

'                    RETURNS PROGRAMME TO THE MAIN MENU AT ALL TIME
on key (1) gosub  1
key (1) on

'                      D E F A U L T     S E T T I N G S

NUMBER = 0
X0     = 1967
Y0     = -350
key off
INSERT= 0
XSTOP = 0
YSTOP = 0
XYSTOP= 0
MULT  = 10
for J = 1 to MAXDIM
  POST$ (J)    = "NOT DEFINED"
  POSCODE$(J) = "00000000000"
next

'                     M A I N    M E N U
1  PORTB = 1
   gosub 54
2  cls
   color 0,7 : print "  MAKE A CHOICE FROM THE MAIN MENU  "
   color 7,0 : print
   color 0,7 : print " 1 ";
   color 7,0 : print "   DEFINE MEASUREMENT POINTS BY USE OF THE REMOTE JOYSTI
   color 0,7 : print " 2 ";
   color 7,0 : print "   DEFINE MEASUREMENT POINTS BY USE OF THE KEYBORD CURSO
   color 0,7 : print " 3 ";
   color 7,0 : print "   INSERT A MEASUREMENT POINT" : print
   color 0,7 : print " 4 ";
   color 7,0 : print "   DELETE A MEASUREMENT POINT" : print
   color 0,7 : print " 5 ";
   color 7,0 : print "   OPTIMIZATION OF THE SCAN SEQUENCE" : print
   color 0,7 : print " 6 ";
   color 7,0 : print "   DISPLAY/DESCRIBE THE MEASUREMENT POINTS" : print
```

```
    color 0,7 : print " 7 ";
    color 7,0 : print "   CHECK COORDINATES BY USE OF THE JOYSTICK" : print
    color 0,7 : print " 8 ";
    color 7,0 : print "   SAVE THE TABLE OF COORDINATES" : print
    color 0,7 : print " 9 ";
    color 7,0 : print "   LOAD THE TABLE OF COORDINATES" : print
    color 0,7 : print " 0 ";
    color 7,0 : print "   PERFORM A TESTRUN" : print
    color 0,7 : print " e ";
    color 7,0 : print "   EXIT  ; NOTE MENU ";
    color 23,0 : print " 6 ";
    color 7,0  : print " ! ";
3   A$ = INKEY$
    if A$ = "1" then
      gosub 4
      goto 1
    elseif A$ = "2" then
      gosub 8
      goto 1
    elseif A$ = "3" then
      goto 19
    elseif A$ = "4" then
      goto 25
    elseif A$ = "5" then
      goto 28
    elseif A$ = "6" then
      goto 32
    elseif A$ = "7" then
      goto 26
    elseif A$ = "8" then
      goto 38
    elseif A$ = "9" then
      goto 42
    elseif A$ = "0" then
      gosub 43
      goto 1
    elseif A$ = "e" or A$ = "E" then
      cls
      print "END OF PROGRAMME"
      end
    else
      goto 3
    end if

    '                   DEFINE MEASUREMENT POINTS BY USE OF THE REMOTE JOYSTICK
    '                   USE THE FUNCTION KEY  F1  TO RETURN TO THE MAIN MENU

    '                   CHECK IF THERE ARE ALREADY MEASUREMENT POINTS DEFINED
4   if NUMBER = 0 then 5
    cls
    color 0,7
    locate 5,20 : print "                                                  "
    locate 6,20 : print "                                                  "
    locate 7,20 : print "     A COORDINATE TABLE IS IN MEMORY              "
    locate 8,20 : print "                                                  "
    locate 9,20 : print "                                                  "
    locate 13,20: print " 1 "; : color 7,0 : print "    OVERWRITE" : color 0,7
    locate 15,20: print " 2 "; : color 7,0 : print "    RETURN TO MAIN MENU"
    V$ = ""
    while V$ <> "1" and V$ <> "2"
      V$ = INKEY$
    wend
    if V$ = "2" then 1
    NUMBER = 0
```

```
5  cls
   IND = 0
   color 0,7
   locate 1,1 : print "
   locate 2,1 : print "                          DEFINE MEASUREMENT POINT COORDINAT
   locate 3,1 : print "
6  locate 5,18: print "                                                  "
   locate 6,18: print "          USE THE REMOTE CONTROL                  "
   locate 7,18: print "                                                  "
   locate 25,1: print " USE FUNCTION KEY F1 TO RETURN TO THE MAIN MENU AFTER Y
   color 7,0
   locate 9,18: print "POSITION" tab (40) "X" tab (50) "Y" ;
   key (1) on

   '                   CHECK IF POSITION IS ACCEPTED AS A MEASUREMENT POINT
   '                   AND DISPLAY MOMENTARILY COORDINATES
7  IRC = inp (1822) and 1
   do while IRC = 1
     IRC = inp (1822) and 1
     XYSTOP = 0
     DAC = 0
     gosub 47
     VWX = VOLT
     DAC = 1
     gosub 47
     VMY = VOLT
     locate 10+IND,21 : print NUMBER tab (38) ; VMX tab (48) ; VMY ;
   loop
'
'    '                   CHECK IF POSITION IS ACCEPTED AS A MEASUREMENT POINT

   wait 1822,1,0
   DAY$ = DATE$

   '                   READ BACK THE SENSOR POSITION
   DAC = 0
   gosub 47
   VX  = VOLT
   DAC = 1
   gosub 47
   VY  = VOLT

   '                   OUTPUT THIS POSITION TO CHECK CORRECTNESS
   DAC = 0
   gosub 48
   DAC = 1
   gosub 48
   if INSERT = 1 then return

   '                   STORE THIS POSITION
   NUMBER    = NUMBER + 1
   X(NUMBER) = VX
   Y(NUMBER) = VY
   IND       = IND + 1
   locate 9+IND,21 : print NUMBER tab (38) ; VX tab (48) ; VY
   if NUMBER = MAXDIM then return 59
   REST = NUMBER MOD 13
   if REST = 0 then 5
   goto 7
   return

   '                   DEFINE MEASUREMENT POINT COORDINATES THROUGH
   '                   KEYBORD CURSOR CONTROL
8  key (1) on
```

```
'                    CHECK IF THERE ARE ALREADY MEASUREMENT POINTS DEFINED
cls
if NUMBER = 0 then 9
color 0,7
locate 5,20 : print "                                                  "
locate 6,20 : print "                                                  "
locate 7,20 : print "     A COORDINATE TABLE IS IN MEMORY               "
locate 8,20 : print "                                                  "
locate 9,20 : print "                                                  "
locate 13,20: print " 1 "; : color 7,0 : print "    OVERWRITE" : color 0,7
locate 15,20: print " 2 "; : color 7,0 : print "    RETURN TO MAIN MENU"
V$ = ""
while V$ <> "1" and V$ <> "2"
  V$ = INKEY$
wend
if  V$ = "2" then 1
NUMBER = 0

'                    CHECK IF THE REMOTE CONTROL IS DISCONNECTED.
9  gosub 55
10 PORTB = 3
   gosub 54              'UNLOCK SERVO SYSTEM
   cls
   color 0,7
   locate 1,1  : print "
   locate 2,1  : print "       DEFINE MEASUREMENT COORDINATES THROUGH THE KEYB
   locate 3,1  : print "
   locate 5,1  : print "USE THE FOLLOWING KEYS"
   color 7,0
   locate 7,1  : print CHR$(24) ; CHR$(25) ; CHR$(26) ; CHR$(27)
   locate 7,5  : print ": CURSOR CONTROLS FOR POSITIONING THE SENSOR HEAD"
   color 0,7
   locate 9,1  : print " a "
   color 7,0
   locate 9,5  : print ": TO ACCEPT AND STORE A MEASUREMENT POINT"
   color 0,7
   locate 11,1 : print " b "
   color 7,0
   locate 11,5 : print ": TO POSITION THE SENSOR AT THE FIRST POSITION AND RET
   color 0,7
   locate 13,1 : print " o "
   color 7,0
   locate 13,5 : print ": TO POSITION THE SENSOR AT THE ORIGIN AND RETURN TO M
   color 0,7
   locate 15,25: print "                              "
   locate 16,25: print "   YOU HAVE CONTROL NOW       "
   locate 17,25: print "                              "
   locate 19,1 : print " MOMENTARILY POSITION "
   locate 19,28: print "  NEW COORDINATES  "
   locate 19,52: print " LAST POSITION STORED "
   color 7,0
   locate 21,5
   print "X" tab(15) "Y" tab(33) "X" tab(43) "Y" tab(53) "POS" tab(61) "X" tab
   locate 25,1 : print "STEPSIZE  ";
   color 0,7 : print " Ctrl-PgUp ";
   color 7,0 : print "=500";
   locate 25,32
   color 0,7 : print " PgUp ";
   color 7,0 : print "=100";
   locate 25,50
   color 0,7 : print " PgDn ";
   color 7,0 : print "=10";
   locate 25,65
```

```
      color 0,7 : print " Ctrl-PgDn ";
      color 7,0 : print "=1";
      PORTB = 3
      gosub 54

      '                    READ THE MOMENTARILY POSITION OF THE SENSOR
      DAC = 0
      gosub 47
      VX  = VOLT
      DAC = 1
      gosub 47
      VY  = VOLT

   11 if INSERT = 1 then
        locate 22,53 : print INS ;
        locate 22,58 : print USING "#####" ; X(INS) ;
        locate 22,66 : print USING "#####" ; Y(INS) ;
      end if
   12 DAC = 0
      gosub 47
      RX  = VOLT
      DAC = 1
      gosub 47
      RY  = VOLT
      locate 22,2  : print USING "#####" ; RX ;
      locate 22,12 : print USING "#####" ; RY ;
   13 locate 22,30 : print USING "#####" ; VX ;
      locate 22,40 : print USING "#####" ; VY ;
      gosub 53
      if XYSTOP = 1 then
        PORTB = 1
        gosub 54
      end if
   14 V$ = ""
      V$ = INKEY$
      if V$ = "" then 11
      if LEN(V$) = 2 then
        V$ = RIGHT$(V$,1)
      else
        goto 16
      end if

      '                   USE THE KEY        TO INCREMENT POSITION WITH STEPSIZE
      '                      PgUp                          100
      '                      PgDn                           10
      '                    Ctrl PgUp                       500
      '                    Ctrl PgDn                         1

      if V$ = CHR$(73) then
        MULT = 10
      elseif V$ = CHR$(81) then
        MULT = 1
      elseif V$ = CHR$(132) then
        MULT = 50
      elseif V$ = CHR$(118) then
        MULT = .1
      end if
      PORTB = 3
      gosub 54
      if V$ = CHR$(72) then
        DAC = 1
        VY  = VY - 10*MULT
        gosub 48
      elseif V$ = CHR$(80) then
```

```
      DAC = 1
      VY  = VY + 10*MULT
      gosub 48
    elseif V$ = CHR$(75) then
      DAC = 0
      VX  = VX + 10*MULT
      gosub 48
    elseif V$ = CHR$(77) then
      DAC = 0
      VX  = VX - 10*MULT
      gosub 48
    end if
    goto 12

16  if V$ <> "a" then 18
17  if NUMBER = MAXDIM then
      return 59
    else
      NUMBER = NUMBER + 1
    end if
    if INSERT <> 1 then 15
    for J = INS+2 to NUMBER
      HX(J)     = X(J-1)
      HY(J)     = Y(J-1)
      H$(J)     = POST$(J-1)
      TETAC(J)  = TETA(J-1)
      AZIMC(J)  = AZIM(J-1)
      HC(J)     = H(J-1)
      RANC(J)   = RAN(J-1)
      EPSC(J)   = EPS(J-1)
      HP$(J)    = POSCODE$(J-1)
      HS$(J)    = SCREEN$(J-1)
      HL$(J)    = ELPOST$(J-1)
      HE$(J)    = ELSCR$(J-1)
    next
    for J = INS+2 to NUMBER
      X(J)         = HX(J)
      Y(J)         = HY(J)
      POST$(J)     = H$(J)
      TETA(J)      = TETAC(J)
      AZIM(J)      = AZIMC(J)
      H(J)         = HC(J)
      RAN(J)       = RANC(J)
      EPS(J)       = EPSC(J)
      POSCODE$(J)= HP$(J)
      SCREEN$(J) = HS$(J)
      ELPOST$(J) = HL$(J)
      ELSCR$(J)  = HE$(J)
    next
    X(INS+1)         = VX
    Y(INS+1)         = VY
    POST$(INS+1)     = ""
    TETA(INS+1)      = 0
    AZIM(INS+1)      = 0
    H(INS+1)         = 0
    RAN(INS+1)       = 0
    EPS(INS+1)       = 0
    POSCODE$(INS+1)= ""
    SCREEN$(INS+1) = ""
    ELPOST$(INS+1) = ""
    ELSCR$(INS+1)  = ""
    INSERT           = 0
    if NUMBER = MAXDIM then
      return 1
```

```
    else
      return
    end if
15  locate 22,53 : print NUMBER ;
    locate 22,58 : print USING "#####" ; VX ;
    locate 22,65 : print USING "#####" ; VY ;
    X(NUMBER) = VX
    Y(NUMBER) = VY
    DAY$      = DATE$
    goto 14

    '                    BRING THE SENSOR TO THE ORIGIN OR THE FIRST POSITION

18  PORTB = 3
    gosub 54
    if V$ = "b" then
      DAC = 0
      VX  = X(1)
      gosub 48
      DAC = 1
      VY  = Y(1)
      gosub 48
      gosub 50
      goto 1
    elseif V$ = "o" then
      DAC = 0
      VX  = X0
      gosub 48
      DAC = 1
      VY  = Y0
      gosub 48
      gosub 50
      goto 1
    end if
    goto 13

    '                    INSERT A NEW MEASUREMENT POINT IN THE TABLE

19  if NUMBER = MAXDIM then 59
    if NUMBER = 0 then
      gosub 46
      goto 1
    end if
    cls
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "                INSERT A MEASUREMENT POINT IN THE EXISTI
    locate 3,1 : print "
    locate 5,1
    color 7,0  : print "    GIVE THE POSITION ";
    color 23,0 : print "AFTER";
    color 7,0  : print " WHICH THE NEW COORDINATES WILL BE INSERTED    "
    locate 9,1
    color 0,7  : print "  DEFAULT  ";
    color 7,0  : print " RETURN TO THE MAIN MENU"
    locate 11,1
    color 0,7  : print "   INPUT   ";
    color 7,0  : input " ",INS
    while INS <= 0 or INS > NUMBER
      color 0,7
      locate 17,18 : print "                                                  "
      locate 18,18 : print "  THE GIVEN MEASUREMENT POINT DOES NOT EXIST  "
      locate 19,18 : print "                                                  "
      locate 20,18 : print "                                                  "
```

```
      locate 21,18 : print "                TRY AGAIN                    "
      locate 22,18 : print "                                             "
      color 7,0
      locate 11,1  : print "                                         "
      color 0,7
      locate 11,1  : print "   INPUT    ";
      color 7,0
      input " ",INS
    wend

21  cls
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "               INSERT A MEASUREMENT POINT IN THE EXISTI
    locate 3,1 : print "
    INSERT = 1
    DAY$   = DATE$
    locate 7,1 : print " 1 "
    locate 7,5
    color 7,0  : print "DIRECT THE SENSOR TO THE POSITION TO BE INSERTED BY THE
    locate 9,1
    color 0,7  : print " 2 "
    locate 9,5
    color 7,0  : print "DIRECT THE SENSOR TO THE POSITION TO BE INSERTED BY THE
    locate 11,1
    color 0,7  : print " 3 "
    locate 11,5
    color 7,0  : print "INSERT ";
    color 0,7  : print " KNOWN ";
    color 7,0  : print " COORDINATES THROUGH THE KEYBORD"
    V$ = ""
    while V$ <> "1" and V$ <> "2" and V$ <> "3"
      V$ = INKEY$
    wend
    if V$ = "2" then
      goto 23
    elseif V$ = "3" then
      goto 24
    end if

    '                  GIVE CONTROL TO THE REMOTE CONTROL
    cls
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "               INSERT A MEASUREMENT POINT IN THE EXISTI
    locate 3,1 : print "
    gosub 6
    gosub 17
22  locate 11,21 : print INS+1
    locate 11,38 : print X(INS+1)
    locate 11,48 : print Y(INS+1)
    goto 22

    '                  GIVE CONTROL TO THE CURSORS, BUT CHECK FIRST IF THE
    '                  REMOTE CONTROL IS DISCONNECTED.
23  cls
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "               INSERT A MEASUREMENT POINT IN THE EXISTI
    locate 3,1 : print "
    color 7,0
    gosub 55
    '                  DIRECT SENSOR TO X(INS),Y(INS) AND GIVE CONTROL
    '                  TO THE CURSORS
```

```
    PORTB = 3
    gosub 54
    DAC   = 0
    VX    = X(INS)
    gosub 48
    DAC   = 1
    VY    = Y(INS)
    gosub 48
    gosub 53
    while XYSTOP <> 1
      locate 25,1
      color 0,7 : print "  MOVING TO POSITION  ";
      color 7,0 : print " " ; INS ; "                    ";
      gosub 53
    wend
    locate 25,1 : print "REACHED POSITION  " ; INS ; "                    ";
    gosub 10
    gosub 17
    goto 1

    '                    GIVE KNOWN COORDINATES
24  cls
    color 0,7
    print "
    print "                      GIVE THE COORDINATES TO BE INSERTED
    print "
    color 7,0
    locate 7,1 : input "X-COORDINATE  :" , VX
    locate 9,1 : input "Y-COORDINATE  :" , VY
    gosub 17
    INSERT = 0
    goto 1

    '                    DELETE A GIVEN MEASUREMENT POINT FROM THE TABLE
25  cls
    if NUMBER = 0 then
      gosub 46
      goto 1
    end if
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "               REMOVE A MEASUREMENT POINT FROM THE EXIST
    locate 3,1 : print "
    locate 7,1 : print "  DEFAULT  ";
    color 7,0
    print " RETURN TO THE MAIN MENU"
    input " ",NB
    while NB <= 0 and NB > NUMBER
      color 0,7
      locate 17,18 : print "                                                  "
      locate 18,18 : print "  THE GIVEN MEASUREMENT POINT DOES NOT EXIST  "
      locate 19,18 : print "                                                  "
      locate 20,18 : print "                                                  "
      locate 21,18 : print "                  TRY AGAIN                       "
      locate 22,18 : print "                                                  "
      color 7,0
      locate 9,1   : print "                                        "
      color 0,7
      locate 9,1   : print "  INPUT  ";
      color 7,0
      input " ",NB
    wend

27  cls
```

```
      color 0,7
      locate 1,1  : print "
      locate 2,1  : print "              REMOVE A MEASUREMENT POINT FROM THE EXIS
      locate 3,1  : print "
      color 7,0
      locate 6,1  : print "POS      X          Y"
      locate 6,25 : print "DESCRIPTION"
      locate 8,1  : print NB
      locate 8,7  : print using "#####" ; X(NB)
      locate 8,17 : print using "#####" ; Y(NB)
      locate 8,25 : print POST$(NB)
      locate 11,3 : print "IS THIS THE CORRECT ONE (Y/N)? :"
      A$ = ""
      while A$ <> "y" and A$ <> "Y" and A$ <> "n" and A$ <> "N"
        A$ = INKEY$
      wend
      if  A$ = "n" or A$ = "N" then 1
      DAY$   = DATE$
      NUMBER = NUMBER - 1
      if  NB = NUMBER + 1 then 1
      for  J = NB to NUMBER
        X(J)       = X(J+1)
        Y(J)       = Y(J+1)
        POST$(J)   = POST$(J+1)
        TETA(J)    = TETA(J+1)
        AZIM(J)    = AZIM(J+1)
        H(J)       = H(J+1)
        RAN(J)     = RAN(J+1)
        EPS(J)     = EPS(J+1)
        POSCODE$(J)= POSCODE$(J+1)
        SCREEN$(J) = SCREEN$(J+1)
        ELPOST$(J) = ELPOST$(J+1)
        ELSCR$(J)  = ELSCR$(J+1)
      next
      goto 1

      '                    OPTIMIZATION OF THE SCAN SEQUENCE
   28 if NUMBER <> 0 then 29
      gosub 46
   29 cls
      color 0,7
      locate 12,26 : print "                          "
      locate 13,26 : print "     OPTIMIZING NOW       "
      locate 14,26 : print "                          "
      color 7,0

      '                    FIRST THE AZIMUTH DIRECTION
      N1 = NUMBER - 1
      for J = 1 to N1
        MAX = X(J)
        for M = J to NUMBER
          if MAX >= X(M) then
            MAX = X(M)
            K = M
          end if
        next M
        swap X(J),X(K)
        swap Y(J),Y(K)
        swap POST$(J),POST$(K)
        swap TETA(J),TETA(K)
        swap AZIM(J),AZIM(K)
        swap H(J),H(K)
        swap RAN(J),RAN(K)
        swap EPS(J),EPS(K)
```

```
      swap POSCODE$(J),POSCODE$(K)
      swap SCREEN$(J),SCREEN$(K)
      swap ELPOST$(J),ELPOST$(K)
      swap ELSCR$(J),ELSCR$(K)
    next J

    '                   IF THERE ARE MORE POINTS, HAVING THE SAME X-COORDINATE,
    '                   THEN OPTIMIZATION IN ELEVATION IS PERFORMED,
    '                   I.E. LOWEST ELEVATION FIRST.
    M = 0
    for J = 1 to N1
      if X(J) <= X(J+1)-.05 or X(J) >= X(J+1)+.05  then
        goto 30
      end if
      M = M + 1
      if M = 1 then K = J
      if J <> N1 then
        goto 31
      end if
30    if M = 0 then 31
      for LL = K to (K+M-1)
        MAX = Y(LL)
        for I = LL to (K+M)
          if MAX >= Y(I) then
            MAX = Y(I)
            L = I
          end if
        next I
        swap Y(L),Y(LL)
        swap POST$(L),POST$(LL)
        swap TETA(L),TETA(LL)
        swap AZIM(L),AZIM(LL)
        swap H(L),H(LL)
        swap RAN(L),RAN(LL)
        swap EPS(L),EPS(LL)
        swap POSCODE$(L),POSCODE$(LL)
        swap SCREEN$(L),SCREEN$(LL)
        swap ELPOST$(L),ELPOST$(LL)
        swap ELSCR$(L),ELSCR$(LL)
      next LL
      M = 0
31  next J
    goto 1

    '                   DISPLAY THE COORDINATES OF THE DEFINED MEASUREMENT
    '                   POINTS
32  key (1) on
    if NUMBER = 0 then
      gosub 46
      goto 1
    end if
    cls
    color 0,7
    locate 1,1  : print "
    locate 2,1  : print "                    DISPLAY AND/OR DESCRIBE MEASUREMENT
    locate 3,1  : print "
    locate 5,1  : print " DISPLAY "
    locate 7,1  : print " 1 "
    locate 9,1  : print " 2 "
    locate 11,1 : print " 3 "
    locate 14,1 : print " INPUT DESCRIPTION FOR "
    locate 16,1 : print " 4 "
    locate 18,1 : print " 5 "
    locate 20,1 : print " 6 "
```

```
   color 7,0
   locate 7,6  : print "ONE POSITION"
   locate 9,6  : print "ALL POSITIONS, ONE BY ONE"
   locate 11,6 : print "SELECTED INFORMATION OF ALL POSITIONS"
   locate 16,6 : print "ONE POSITION"
   locate 18,6 : print "ALL POSITIONS"
   locate 20,6 : print "EACH POSITION AND ";
   color 0,7   : print " M O V E ";
   color 7,0   : print " SENSOR HEAD"
   locate 24,1 : print "CHOOSE OPTION" TAB(45) "F1 = RETURN TO MAIN MENU";
33 Z$ = INKEY$
   if Z$ = "1" then
     goto 41
   elseif Z$ = "2" then
     goto 40
   elseif Z$ = "3" then
     goto 61
   elseif Z$ = "4" then
     goto 37
   elseif Z$ = "5" then
     goto 35
   elseif Z$ = "6" then
     goto 34
   end if
   goto 33

   '                    INPUT INFORMATION FOR EACH POSITION
   '                      WITH MOVING TO POSITION
34 cls
   gosub 55             'CHECK IF REMOTE CONTROL IS DISCONNECTED

   '                    INPUT INFORMATION FOR EACH POSITION
   '                      WITHOUT MOVING TO POSITION
35 for  NB = 1 to NUMBER
     if Z$ = "5" then 36
     PORTB = 3
     gosub 54
     VX    = X(NB)
     DAC   = 0
     gosub 48
     VY    = Y(NB)
     DAC   = 1
     gosub 48
     gosub 53
     while XYSTOP <> 1
       locate 25,1 : print "  MOVING TO POSITION  ";
       color 7,0   : print " " ; NB ;
       gosub 53
     wend
     PORTB = 1
     gosub 54
     locate 25,1  : print "
     locate 25,1  : print "REACHED POSITION  " ; NB ;
     color 0,7
     locate 25,35 : print "  GIVE INFORMATION NOW  ";
     color 7,0
36   cls
     gosub 56
     gosub 58
   next NB
   goto 1

   '                    INPUT INFORMATION FOR ONE POSITION
37 cls
```

```
    color 0,7
    print "
    print "                    INPUT INFORMATION FOR ONE POSITION
    print "
    locate 7,1 : print " DEFAULT ";
    color 7,0  : print " RETURN TO THE MAIN MENU"
    color 0,7  : locate 9,1 : print "  INPUT  ";
    color 7,0  : input " " , NB
    while NB <= 0 or NB > NUMBER
      color 0,7
      locate 17,18 : print "                                                  "
      locate 18,18 : print "  THE GIVEN MEASUREMENT POINT DOES NOT EXIST  "
      locate 19,18 : print "                                                  "
      locate 20,18 : print "                                                  "
      locate 21,18 : print "                  TRY AGAIN                       "
      locate 22,18 : print "                                                  "
      color 7,0
      locate 9,1   : print "                                        "
      color 0,7
      locate 9,1   : print "  INPUT  ";
      color 7,0
      input " " , NB
    wend

 39 cls
    gosub 56           'DISPLAY
    gosub 58           'DESCRIBE
    goto 32

    '                   DISPLAY ALL MEASUREMENT POINTS
    '                   - ALL INFORMATION
    '                   - SELECTED INFORMATION
 40 for NB = 1 to NUMBER
      gosub 56
      locate 24,1  : print "HIT A KEY TO CONTINUE";
      locate 24,45 : print "F1  = RETURN TO MAIN MENU";
      do
      loop while INKEY$ = ""
    next NB
    goto 32

    '                   DISPLAY INFORMATION FOR ONE POSITION
 41 cls
    color 0,7
    print "
    print "                    DISPLAY INFORMATION FOR ONE POSITION
    print "
    locate 7,1 : print " DEFAULT ";
    color 7,0  : print " RETURN TO THE MAIN MENU"
    color 0,7  : locate 9,1 : print "  INPUT  ";
    color 7,0  : input " " , NB
    while NB <= 0 or NB > NUMBER
      color 0,7
      locate 17,18 : print "                                                  "
      locate 18,18 : print "  THE GIVEN MEASUREMENT POINT DOES NOT EXIST  "
      locate 19,18 : print "                                                  "
      locate 20,18 : print "                                                  "
      locate 21,18 : print "                  TRY AGAIN                       "
      locate 22,18 : print "                                                  "
      color 7,0
      locate 9,1   : print "                                        "
      color 0,7
      locate 9,1   : print "  INPUT  ";
      color 7,0
```

```
      input " " , NB
    wend
    gosub 56
    locate 25,1 : print "HIT A KEY TO CONTINUE";
    locate 25,45: print "F1 = RETURN TO MAIN MENU";
    locate 23,1
    do
    loop while INKEY$ = ""
    goto 32

    '                    WRITE THE COORDINATE TABLE TO DISK
38  if NUMBER = 0 then
      gosub 46
      goto 1
    end if
    cls
    color 0,7
    locate 1,1  : print "
    locate 2,1  : print "       CATALOG THE MEASUREMENT POINT COORDINATE FILE ON
    locate 3,1  : print "
    locate 5,18 : print "                                                      "
    locate 6,18 : print "        PUT AN EMPTY DISKETTE IN DRIVE B              "
    locate 7,18 : print "                                                      "
    DMY$ = FN DAG$(DATE$)
    FIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "P"
    locate 10,1 : print " FORMAT  "
    locate 12,1 : print " DEFAULT "
    locate 15,1 : print "  INPUT  "
    color 7,0
    locate 10,13 : print " MMMDDYYP    "
    locate 12,13 : print " ";FIL$
    locate 15,13 : line input " ",DESC$
    if DESC$ = "" then DESC$ = FIL$
    cls
    color 0,7
    locate 8,20  : print "                                              "
    locate 9,20  : print "                                              "
    locate 10,20 : print "        WRITING FILE TO DISKETTE              "
    locate 11,20 : print "                                              "
    locate 12,20 : print "                                              "
    color 7,0
    open "o",#1,"B:"+DESC$
      print #1,USING "###" ; NUMBER
      print #1,USING "\        \" ; DAY$
      for J = 1 to NUMBER
        print #1,USING "\                              \";POST$(J)
        print #1,USING "\                            \" ; ELPOST$(J)
        print #1,USING "+####.##" ; X(J)
        print #1,USING "+####.##" ; Y(J)
        print #1,USING "+####.##" ; TETA(J)
        print #1,USING "+####.##" ; AZIM(J)
        print #1,USING "+####.##" ; H(J)
        print #1,USING "+####.##" ; RAN(J)
        print #1,USING "+####.##" ; EPS(J)
        print #1,USING "\                                 \" ; SCREEN$(J)
        print #1,USING "\                               \" ; ELSCR$(J)
        print #1,USING "\                     \" ; POSCODE$(J)
      next J
    close #1
    goto 1

    '                    LOAD  THE COORDINATE TABLE FROM DISK

    '                    CHECK IF THERE ARE ALREADY MEASUREMENT POINTS DEFINED
```

```
42 cls
   if NUMBER = 0 then 20
   cls
   color 0,7
   locate 5,20 : print "                                                "
   locate 6,20 : print "                                                "
   locate 7,20 : print "     A COORDINATE TABLE IS IN MEMORY             "
   locate 8,20 : print "                                                "
   locate 9,20 : print "                                                "
   locate 13,20: print " 1 "; : color 7,0 : print "    OVERWRITE" : color 0,7
   locate 15,20: print " 2 "; : color 7,0 : print "    RETURN TO MAIN MENU
   V$ = ""
   while V$ <> "1" and V$ <> "2"
     V$ = INKEY$
   wend
   if V$ = "2" then 1
   NUMBER = 0

20 cls
   color 0,7
   print "
   print "  GIVE THE NAME OF THE COORDINATE FILENAME (ON DRIVE B) TO LOAD INTO
   print "
   DMY$ = FN DAG$(DATE$)
   FIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "P"
   locate 5,1 : print " FORMAT  "
   locate 7,1 : print " DEFAULT "
   locate 9,1 : print "  INPUT  "
   color 7,0
   locate 5,13 : print " MMMDDYYP    "
   locate 7,13 : print " ";FIL$
   locate 9,13 : line input " ",DESC$
   if DESC$ = "" then DESC$ = FIL$
   cls
   color 0,7
   locate 8,20  : print "                                               "
   locate 9,20  : print "                                               "
   locate 10,20 : print "     LOADING THE FILE INTO MEMORY              "
   locate 11,20 : print "                                               "
   locate 12,20 : print "                                               "
   color 7,0
   open "I",#1,"B:"+DESC$
     input #1 , NUMBER
     input #1 , DAY$
     for J = 1 to NUMBER
       input #1 , POST$(J)
       input #1 , ELPOST$(J)
       input #1 , X(J)
       input #1 , Y(J)
       input #1 , TETA(J)
       input #1 , AZIM(J)
       input #1 , H(J)
       input #1 , RAN(J)
       input #1 , EPS(J)
       input #1 , SCREEN$(J)
       input #1 , ELSCR$(J)
       input #1 , POSCODE$(J)
     next J
   close #1
   goto 1

   '                    CHECK POSITIONS BY USE OF THE JOYSTICK

26 if NUMBER = 0 then
```

```
      gosub 46
      goto 1
    end if
    cls
    color 0,7
    locate 1,1  : print "
    locate 2,1  : print "                CHECK MEASUREMENT POINT COORDINATES WIT
    locate 3,1  : print "
    color 7,0
    locate 6,1  : print "PRESS THE ACCEPT BUTTON ON THE REMOTE CONTROL TO GO TH
    color 0,7
    locate 8,1  : print " HOLD THE BUTTON DOWN ";
    color 7,0
    print " TILL THE SENSOR HAS REACHED ITS POSITION."
    locate 12,1 : print "TO GO TO THE NEXT POSITION ";
    color 0,7
    print " PRESS AND HOLD DOWN ";
    color 7,0
    print " THE BUTTON AGAIN"

    for J = 1 to NUMBER

      '                  CHECK IF FUNCTION KEY F1 WAS USED
      key(1) on
      color 0,7
      locate 25,1
      print "    USE FUNCTION KEY F1 TO RETURN TO THE MAIN MENU AFTER YOU ARE F
      color 7,0

      '                  CHECK IF ACCEPT BUTTON WAS PRESSED
      wait 1822,1,0

      '                  OUTPUT THIS POSITION TO CHECK CORRECTNESS
      PORTB = 3
      gosub 54
      DAC   = 0
      VX    = X(J)
      gosub 48
      DAC   = 1
      VY    = Y(J)
      gosub 48
      gosub 53
      while XYSTOP <> 1
        gosub 53
      wend
      PORTB = 1
      gosub 54
    next J
    goto 1

    '                  PERFORM A TESTRUN  TO TEST POSITION ACCURACY ONLY
43  cls
    if NUMBER = 0 then
      gosub 46
      goto 1
    end if
    '                  CHECK IF THE REMOTE CONTROL IS DISCONNECTED.
    gosub 55

    '                  POSITION THE SENSOR AT THE FIRST POSITION

    PORTB = 3
    gosub 54           'UNLOCK SERVO SYSTEM
    DAC   = 0
```

```
      VX    = X(1)
      gosub 48
      DAC   = 1
      VY    = Y(1)
      gosub 48
      gosub 50

      cls
      color 0,7
      print "
      locate 2,1   : print "                PERFORM A TESTRUN, TO TEST POSITION AC
      locate 3,1   : print "
      locate 5,1   : print "USE THE FOLLOWING KEY"
      locate 7,1   : print " n "
      locate 9,1   : print " b "
      locate 11,1  : print " o "
      locate 13,27 : print "                          "
      locate 14,27 : print "  YOU HAVE CONTROL NOW  "
      locate 15,27 : print "                          "
      locate 17,1  : print "MOMENTARILY POSITION"
      locate 17,26 : print "   NEXT POSITION    "
      locate 17,52 : print "   LAST POSITION    "

      color 7,0
      locate 7,5   : print ": TO GO TO THE NEXT POSITION"
      locate 9,5   : print ": TO POSITION THE SENSOR AT THE FIRST POSITION"
      locate 11,5  : print ": TO POSITION THE SENSOR AT THE ORIGIN"
      locate 19,5  : print "X" tab(15) "Y" tab(27) "POS" tab(35) "X" tab(43) "Y"
      locate 19,53 : print "POS" tab(61) "X" tab(69) "Y"
      locate 20,53 : print NUMBER
      locate 20,58 : print USING "#####" ; X(NUMBER)
      locate 20,66 : print USING "#####" ; Y(NUMBER)
      for J = 1 to NUMBER
        N = J + 1
        if J = NUMBER then N = NUMBER
44    gosub 53
      locate 20,2  : print USING "#####" ; VMX
      locate 20,12 : print USING "#####" ; VMY
      locate 20,27 : print J
      locate 20,32 : print USING "#####" ; X(J)
      locate 20,40 : print USING "#####" ; Y(J)
      if XYSTOP <> 1 then 44
      locate 20,27 : print N
      locate 20,32 : print USING "#####" ; X(N)
      locate 20,40 : print USING "#####" ; Y(N)
      PORTB = 1
      gosub 54

45    color 0,7
      locate 23,35 : print "              DESCRIPTION              "
      locate 25,1  : print "  REACHED POSITION  ";
      color 7,0
      print "  " ; J"      " ;
      locate 25,35 : print POST$(J);
      V$ = ""
      while V$ <> "n" and V$ <> "b" and V$ <> "o"
        V$ = INKEY$
      wend
      PORTB = 3
      gosub 54
      if V$ <> "n" then
        DAC = 0
        if V$ = "b" then
          VX  = X(1)
```

```
        gosub 48
        DAC = 1
        VY  = Y(1)
        gosub 48
        gosub 50
        V$ = ""
        goto 1
      elseif V$ = "o" then
        VX  = X0
        gosub 48
        DAC = 1
        VY  = Y0
        gosub 48
        gosub 50
        V$ = ""
        goto 1
      end if
    end if
    DAC = 1
    VY  = Y(N)
    gosub 48
    DAC = 0
    VX  = X(N)
    gosub 48
    color 0,7
    locate 25,1 : print "  MOVING TO THE NEXT POSITION  ";
    color 7,0
    print "                                              ";
    V$ = ""
  next J
  return

  '                    S U B R O U T I N E S

  '                    DISPLAY THAT NO POINTS ARE DEFINED
46 cls
  color 0,7
  locate 5,20  : print "                                          "
  locate 6,20  : print "                                          "
  locate 7,20  : print "      THERE IS NO COORDINATE TABLE        "
  locate 8,20  : print "                                          "
  locate 9,20  : print "                                          "
  locate 13,20 : print "HIT ANY KEY TO CONTINUE"
  locate 23,1
  do
  loop while INKEY$ = ""
  return

  '                    READ A SPECIFIED CHANNEL FROM THE A/D CONVERTER

  '                    PARAMETERS USED:
  '                    -- DAC          A/D CHANNEL TO BE READ
  '                    -- VOLT         VOLTAGE ON THE CHANNEL
47 ADRES = 1808
  out ADRES +4,128                                  'AUTO INCREMENTING OFF
  out ADRES +5,DAC                                  'SELECT A/D CHANNEL
  out ADRES +6,0                                    'START CONVERSION
  while INP(ADRES + 4) < 128
  wend                                              'WAIT FOR CONVERSION READY
  LOW = INP(ADRES + 5)                              'LOWER BYTE
  HIGH = INP(ADRES + 6)                             'HIGHER BYTE
  VOLT = 256*HIGH + LOW
  if VOLT > 32767 then VOLT = VOLT - 65536!         'RANGE  -10 TO +10 VOLT
  if DAC = 0 then VOLT = VOLT*2.100428# - 2040!     'CONVERSION TO BITS
```

```
    if DAC = 1 then VOLT = VOLT*1.994169# - 2040!
    return

    '                    OUTPUT A VOLTAGE TO A SPECIFIED CHANNEL
    '                    PARAMETERS USED:
    '                    -- DAC = 0       OUTPUT CHANNEL FOR AZIMUTH
    '                    -- DAC = 1       OUTPUT CHANNEL FOR ELEVATION
    '                    -- VY            VOLTAGE ON ELEVATION CHANNEL
    '                    -- VX            VOLTAGE ON AZIMUTH CHANNEL
 48 ADRES  = 1808
    if DAC = 0 then 49
    VY = INT (VY)
    if VY > 2047  then VY =  2047                    'UPPER LIMIT ELEVATION DIRE
    if VY < -1910 then VY = -1910                    'LOWER LIMIT
    HIGHY = INT (VY/256)                             'HIGH VY-BYTE
    LOWY  = VY - 256*HIGHY
    if HIGHY < 0 then HIGHY = HIGHY + 16
    out ADRES+1 , HIGHY                              'ACTIVATE ELEVATION MOTOR
    out ADRES   , LOWY
    return

 49 VX   = INT (VX)
    if VX > 2040  then VX =  2040                    'UPPER LIMIT IN AZIMUTH
    if VX < -2010 then VX = -2010                    'LOWER LIMIT
    HIGHX = INT (VX/256)                             'HIGH VX-BYTE
    LOWX  = VX - 256*HIGHX
    if HIGHX < 0 then HIGHX = HIGHX + 16
    out ADRES+3 , HIGHX                              'ACTIVATE AZIMUTH MOTOR
    out ADRES+2 , LOWX
    return

    '                   DISPLAY THE MOMENTARILY POSITION OF THE SENSOR HEAD
    '                   PARAMETERS USED:
    '                   -- VMX           MOMENTARILY X - COORDINATE
    '                   -- VMY           MOMENTARILY Y - COORDINATE
    '                   -- VX  AND  VY
 50 gosub 53
    if XYSTOP = 1 then
      XYSTOP = 0
      return
    end if
    cls
    color 0,7
    locate 1,29 : print "                          "
    locate 2,29 : print " SENSOR HEAD IS MOVING "
    locate 3,29 : print "                          "
    locate 5,10 : print "                        "
    locate 5,50 : print "                       "
    locate 6,10 : print " MOMENTARILY POSITION "
    locate 6,50 : print "    NEXT POSITION     "
    locate 7,10 : print "                        "
    locate 7,50 : print "                       "
    color 7,0
    locate 9,13 : print "X            Y"
    locate 9,52 : print "X             Y"
    locate 17,10
    color 0,7 : print " s ";
    color 7,0 : print "  : STOP THE SENSOR HEAD AND RETURN TO THE MAIN MENU"

 51 gosub 53
    if XYSTOP = 1 then
      XYSTOP = 0
      return
    end if
```

```
    V$  = INKEY$
    if V$ <> "s" then 52
    DAC = 0
    VX  = VMX
    gosub 48
    DAC = 1
    VY  = VMY
    gosub 48
    goto 1
52  locate 10,10 : print USING "#####" ; VMX
    locate 10,23 : print USING "#####" ; VMY
    locate 10,50 : print USING "#####" ; VX
    locate 10,63 : print USING "#####" ; VY
    goto 51
    return

    '                   TEST IF THE SENSOR HEAD HAS REACHED THE GIVEN POSITION
    '                   PARAMETERS USED:
    '                   -- VX  AND  VY
    '                   -- VMX AND  VMY
53  XYSTOP = 0
    DAC    = 0
    gosub 47
    VMX    = VOLT
    DAC    = 1
    gosub 47
    VMY    = VOLT
    if VMX > VX-5 and VMX < VX+5 then XSTOP = 1
    if VMY > VY-5 and VMY < VY+5 then YSTOP = 1
    if XSTOP = 1 and YSTOP = 1 then
      XSTOP  = 0
      YSTOP  = 0
      XYSTOP = 1
    end if
    return

    '                   LOCK- AND UNLOCK THE SERVO SYSTEM

    '                   PORTB = 0   START MIRROR, OPEN THE HATCH AND LOCK
    '                   PORTB = 1   STOP MIRROR, CLOSE THE HATCH AND LOCK
    '                   PORTB = 2   START MIRROR, OPEN THE HATCH AND UNLOCK
    '                   PORTB = 3   STOP MIRROR, CLOSE THE HATCH AND UNLOCK
54  out 1823,145
    out 1821,PORTB
    return

    '                   CHECK IF THE REMOTE CONTROL IS DISCONNECTED,
    '                   BY READING THE MIRROR POSITION TWICE. IF POSITION
    '                   IS NOT CHANGED THEN REMOTE CONTROL IS DISCONNECTED.
55  ROT0 = INP(1820)
    for TELL = 1 to 2
      PP$ = TIME$
      while TIME$ <= PP$
      wend
    next TELL
    if (INP(1820)-ROT0) = 0 then return
    color 0,7
    locate 17,18 : print "                                              "
    locate 18,18 : print "         DISCONNECT THE REMOTE CONTROL        "
    locate 19,18 : print "                                              "
    color 7,0
    locate 22,18 : print "HIT ANY KEY WHEN READY"
    locate 23,1
    do
```

```
    loop while INKEY$ = ""
    return

    '                    DISPLAY POSITION INFORMATION
 56 cls
    color 0,7
    locate 1,1  : print "
    locate 2,1  : print "                          DISPLAY POSITION INFORMATION
    locate 3,1  : print "
    locate 5,60 : print " DATE "; : color 7,0 :   print " " ; DAY$
    locate 5,1  : print "POSITION "              TAB(25) ":" ; NB
    locate 7,1  : print "X COORDINATE"           TAB(25) ":" ; X (NB)
    locate 8,1  : print "Y COORDINATE"           TAB(25) ":" ; Y (NB)
    locate 10,1 : print "GENERAL DESCRIPTION"    TAB(25) ":" ; POST$ (NB)
    locate 11,1 : print "ELEMENT SPECIFICATION" TAB(25) ":" ; ELPOST$ (NB)
    locate 12,1 : print "ELEVATION ANGLE (DEG)" TAB(25) ":" ; TETA (NB)
    locate 13,1 : print "AZIMUTH ANGLE (DEG)"    TAB(25) ":" ; AZIM (NB)
    locate 14,1 : print "HEIGHT (CM)"            TAB(25) ":" ; H (NB)
    locate 15,1 : print "RANGE (M)"              TAB(25) ":" ; RAN (NB)
    locate 16,1 : print "EMISSIVITY"             TAB(25) ":" ; EPS (NB)
    locate 17,1 : print "BACKGROUND DESCRIPTION"TAB(25) ":" ; SCREEN$ (NB)
    locate 18,1 : print "ELEMENT SPECIFICATION" TAB(25) ":" ; ELSCR$ (NB)
    locate 19,1 : print "POSITION CODE"          TAB(25) ":" ; POSCODE$ (NB)
    if Z$ = "1" or Z$ > "3" then
      if Z$ = "5" or Z$ = "6" then
        locate 21,1 : print "READY TO CONTINUE  (Y) ?             " ;
        P$ = ""
        while P$ <> "y" and P$ <> "Y"
          P$ = INKEY$
        wend
      else
        locate 21,1 : print "IS THIS THE CORRECT POSITION ? (y/n)" ;
        P$ = ""
        while P$ <> "y" and P$ <> "Y" and P$ <> "n" and P$ <> "N"
          P$ = INKEY$
        wend
        if P$ = "n" or P$ = "N" then
          return 37
        end if
      end if
    end if
    return

    '                    INPUT POSITION INFORMATION
 58 cls
    color 0,7
    print "
    print "                              SELECT POSITION DESCRIPTION
    print "
    color 7,0
    locate 5,1  : print "CODE  DESCRIPTION         "
    locate 5,42 : print "CODE  DESCRIPTION         "
    for B = 1 to 42 step 41
      for C = 7 to 21
        locate C,B
        if B = 42 then
          print CODE$ (C+9)
        else
          print CODE$ (C-6)
        end if
      next C
    next B
    print
```

```
        print "CURRENT POSITION IS " ; NB; " : " ; POST$ (NB)
        input;"TYPE CODE TO SELECT A NEW DESCRIPTION :   " , CP
        for C = 1 to 30
          if CP = VAL(LEFT$(CODE$(C),2)) then
            POST$ (NB) = RIGHT$(CODE$(C),19)
          end if
        next C

        cls
        color 0,7
        print "
        print "                                SELECT ELEMENT DESCRIPTION
        print "
        color 7,0
        locate 5,1  : print "CODE  DESCRIPTION        "
        print
        select case CP
          case 1,2,21,22
            for T = 5 to 9
              print ELCODE$ (T)
            next T
          case 3 to 6
            for T = 1 to 4
              print ELCODE$ (T)
            next T
            print ELCODE$ (9)
          case 9
            for T = 1 to 10
              print ELCODE$ (T)
            next T
          case 11 to 16,19,23,29,31 to 33,39,41 to 45,49,91,92
            for T = 8 to 10
              print ELCODE$ (T)
            next T
          case else
            print "YOU CHOSE A NON-EXSISTING POSITION DESCRIPTION"
            print "HIT ANY KEY TO RETURN TO PREVIOUS MENU"
            do
            loop while INKEY$ = ""
            goto 58
        end select
    65  locate 23,1
        print "CURRENT POSITION IS " ; NB ; " : " ; POST$ (NB)
        input;"TYPE CODE TO SELECT ELEMENT DESCRIPTION :   ",CE
        select case CP
          case 1,2,21,22
            if CE<5 or CE>9 then 65
          case 3 to 6
            if CE<>1 and CE<>2 and CE<>3 and CE<>4 and CE<>9 then 65
          case 9
            if CE<0 or CE>9 then 65
          case 11 to 16,19,23,29,31 to 33,39,41 to 45,49,91,92
            if CE<>0 and CE<>8 and CE<>9 then 65
        end select
        for C = 1 to 10
          if CE = VAL(LEFT$(ELCODE$(C),1)) then
            ELPOST$ (NB) = RIGHT$(ELCODE$(C),19)
          end if
        next C

    63  cls
        color 0,7
        print "
        print "                                SELECT " ;
```

```
color 16,7  : print "BACKGROUND" ;
color 0,7   : print " DESCRIPTION                    "
print "
color 7,0
locate 5,1  : print "CODE  DESCRIPTION        "
locate 5,42 : print "CODE  DESCRIPTION        "
for B = 1 to 42 step 41
  for C = 7 to 21
    locate C,B
    if B = 42 then
      print CODE$ (C+9)
    else
      print CODE$ (C-6)
    end if
  next C
next B
print
print "CURRENT POSITION IS " ; NB ; " : " ; POST$ (NB)
print "TYPE CODE TO SELECT ";
color 23,0  : print "BACKGROUND" ;
color 7,0   : input;" DESCRIPTION :   ", CS
for C = 1 to 30
  if CS = VAL(LEFT$(CODE$(C),2)) then
    SCREEN$ (NB) = RIGHT$(CODE$(C),19)
  end if
next C

cls
color 0,7
print "
print "                          SELECT " ;
color 16,7  : print "BACKGROUND" ;
color 0,7   : print " ELEMENT DESCRIPTION                    "
print "
color 7,0
locate 5,1  : print "CODE  DESCRIPTION        "
print
select case CS
  case 1,2,21,22
    for T = 5 to 9
      print ELCODE$ (T)
    next T
  case 3 to 6
    for T = 1 to 4
      print ELCODE$ (T)
    next T
    print ELCODE$ (9)
  case 9
    for T = 1 to 10
      print ELCODE$ (T)
    next T
  case 11 to 16,19,23,29,31 to 33,39,41 to 45,49,91,92
    for T = 8 to 10
      print ELCODE$ (T)
    next T
  case else
    print "YOU CHOSE A NON-EXSISTING BACKGROUND DESCRIPTION"
    print "HIT ANY KEY TO RETURN TO PREVIOUS MENU"
    do
    loop while INKEY$ = ""
    goto 63
end select
64 locate 23,1
print "CURRENT BACKGROUND IS " ; NB ; " : " ; SCREEN$ (NB)
```

```
print "TYPE CODE TO SELECT ";
color 23,0  : print "BACKGROUND" ;
color 7,0   : input;" ELEMENT DESCRIPTION :   ", CB
select case CS
  case 1,2,21,22
    if CB<5 or CB>9 then 64
  case 3 to 6
    if CB<>1 and CB<>2 and CB<>3 and CB<>4 and CB<>9 then 64
  case 9
    if CB<0 or CB>9 then 64
  case 11 to 16,19,23,29,31 to 33,39,41 to 45,49,91,92
    if CB<>0 and CB<>8 and CB>9 then 64
end select
for C = 1 to 10
  if CB = VAL(LEFT$(ELCODE$(C),1)) then
    ELSCR$ (NB) = RIGHT$(ELCODE$(C),19)
  end if
next C

cls
color 0,7
locate 1,1   : print "
locate 2,1   : print "                              INPUT POSITION INFORMATION
locate 3,1   : print "
locate 5,60  : print " DATE " ; : color 7,0 : print " " ; DAY$
locate 5,1   : print "POSITION "               TAB(25) ":" ; NB
locate 7,1   : print "X COORDINATE"           TAB(25) ":" ; X (NB)
locate 8,1   : print "Y COORDINATE"           TAB(25) ":" ; Y (NB)
locate 10,1  : print "GENERAL DESCRIPTION"    TAB(25) ":" ; POST$ (NB)
locate 11,1  : print "ELEMENT SPECIFICATION"  TAB(25) ":" ; ELPOST$ (NB)
locate 12,1  : print "ELEVATION ANGLE (DEG)"  TAB(25) ":" TAB(60) "NOW: " T
locate 13,1  : print "AZIMUTH ANGLE (DEG)"    TAB(25) ":" TAB(60) "NOW: " A
locate 14,1  : print "HEIGHT (CM)"            TAB(25) ":" TAB(60) "NOW: " H
locate 15,1  : print "RANGE (M)"              TAB(25) ":" TAB(60) "NOW: " R
locate 16,1  : print "EMISSIVITY"             TAB(25) ":" TAB(60) "NOW: " E
locate 17,1  : print "BACKGROUND DESCRIPTION" TAB(25) ":" ; SCREEN$ (NB)
locate 18,1  : print "ELEMENT SPECIFICATION"  TAB(25) ":" ; ELSCR$ (NB)
locate 19,1  : print "POSITION CODE"          TAB(25) ":" ; POSCODE$ (NB)
locate 12,25 : input ":" , TETA (NB)
locate 13,25 : input ":" , AZIM (NB)
locate 14,25 : input ":" , H (NB)
do
  locate 15,25 : input ":" , RAN (NB)
loop while RAN (NB) < 0
do
  locate 16,25 : input ":" , EPS (NB)
loop while EPS (NB) < 0 or EPS (NB) > 1

'                  PUT DESCRIPTION OF ELEMENTS INTO AN ELEMENT CODE

while TETA (NB) >= 180
  TETA (NB) = TETA (NB) - 180
wend
while TETA (NB) < 0
  TETA (NB) = TETA (NB) + 180
wend
select case TETA (NB)
  case 0
    CTETA = 1
  case 1 to 45
    CTETA = 2
  case 46 to 89
    CTETA = 3
  case 90
```

```
    CTETA = 4
  case else
    CTETA = 9
end select

while AZIM (NB) >= 360
  AZIM (NB) = AZIM (NB) - 360
wend
while AZIM (NB) < 0
  AZIM (NB) = AZIM (NB) + 360
wend
select case AZIM (NB)
  case is < 45
    CAZIM = 1
  case 45 to 89
    CAZIM = 2
  case 90 to 134
    CAZIM = 3
  case 135 to 179
    CAZIM = 4
  case 180 to 224
    CAZIM = 5
  case 225 to 269
    CAZIM = 6
  case 270 to 314
    CAZIM = 7
  case 315 to 359
    CAZIM = 8
  case else
    CAZIM = 9
end select


select case H (NB)
  case is < 1
    CH = 1
  case 1 to 4
    CH = 2
  case 5 to 9
    CH = 3
  case 10 to 24
    CH = 4
  case 25 to 49
    CH = 5
  case 50 to 99
    CH = 6
  case 100 to 199
    CH = 7
  case 200 to 499
    CH = 8
  case else
    CH = 9
end select


select case RAN (NB)
  case is < 20              'FOV:    (m)
    CRAN = 1                '     0    - .35
  case 20 to 49             '
    CRAN = 2                '     .35 - .875
  case 50 to 99             '
    CRAN = 3                '         - 1.75
  case 100 to 199           '
    CRAN = 4                '         - 3.50
```

```
  case 200 to 299          '
    CRAN = 5               '        - 5.25
  case 300 to 499          '
    CRAN = 6               '        - 8.75
  case 500 to 999          '
    CRAN = 7               '        -17.50
  case is > 999            '
    CRAN = 8               '   17.50- ...
  case else                '
    CRAN = 9               '        -
end select


if EPS (NB) < 0.2 then
  CEPS = 1
elseif EPS (NB) = 1 then
  CEPS = 9
else
  CEPS = INT (10 * (EPS (NB) + 0.0001))
end if

'THE ELEMENTS OF THE POSITION CODE ARE KNOWN BY NOW AND AVAILABLE
'AS     :   CP-CE-CTETA-CAZIM-CH-CRAN-CEPS-CS-CB
'WITH   :    2+ 1+  1  +  1  + 1+  1 +  1 + 2+ 1   CHARACTERS
'THIS CODE OF 11 NUMBERS WILL BE READ INTO A STRING : POSCODE$

CP$    = FNNUM$ (CP)
if LEN (CP$) <> 2 then
  CP$ = "0" + CP$
end if
CE$    = FNNUM$ (CE)
CTETA$ = FNNUM$ (CTETA)
CAZIM$ = FNNUM$ (CAZIM)
CH$    = FNNUM$ (CH)
CRAN$  = FNNUM$ (CRAN)
CEPS$  = FNNUM$ (CEPS)
CS$    = FNNUM$ (CS)
if LEN (CS$) <> 2 then
  CS$ = "0" + CS$
end if
CB$    = FNNUM$ (CB)
POSCODE$ (NB) = CP$ + CE$ + CTETA$ + CAZIM$ + CH$ + CRAN$ + CEPS$ + CS$ + C

cls
color 0,7
locate 1,1   : print "
locate 2,1   : print "                          DISPLAY POSITION INFORMATION
locate 3,1   : print "
locate 5,60  : print " DATE " ; : color 7,0 : print " " ; DAY$
locate 5,1   : print "POSITION "              TAB(25) ":" ; NB
locate 7,1   : print "X COORDINATE"           TAB(25) ":" ; X (NB)
locate 8,1   : print "Y COORDINATE"           TAB(25) ":" ; Y (NB)
locate 10,1  : print "GENERAL DESCRIPTION"    TAB(25) ":" ; POST$ (NB)
locate 11,1  : print "ELEMENT SPECIFICATION"  TAB(25) ":" ; ELPOST$ (NB)
locate 12,1  : print "ELEVATION ANGLE (DEG)"  TAB(25) ":" ; TETA (NB)
locate 13,1  : print "AZIMUTH ANGLE (DEG)"    TAB(25) ":" ; AZIM (NB)
locate 14,1  : print "HEIGHT (CM)"           TAB(25) ":" ; H (NB)
locate 15,1  : print "RANGE (M)"             TAB(25) ":" ; RAN (NB)
locate 16,1  : print "EMISSIVITY"            TAB(25) ":" ; EPS (NB)
locate 17,1  : print "BACKGROUND DESCRIPTION" TAB(25) ":" ; SCREEN$ (NB)
locate 18,1  : print "ELEMENT SPECIFICATION"  TAB(25) ":" ; ELSCR$ (NB)
locate 19,1  : print "POSITION CODE"          TAB(25) ":" ; POSCODE$ (NB)
locate 23,1  : print "HIT ANY KEY TO CONTINUE"
do
```

```
    loop while INKEY$ = ""
    if Z$ = "5" or Z$ = "6" then
      return
    else
      goto 32
    end if

    '                    DISPLAY THAT MAXIMUM OF 'MAXDIM' POSITIONS IS REACHED
 59 cls
    color 0,7
    locate 5,20  : print "                                                  "
    locate 6,20  : print "                                                  "
    locate 7,20  : print "    MAXIMUM OF ";MAXDIM;" POSITIONS IS REACHED    "
    locate 8,20  : print "                                                  "
    locate 9,20  : print "                                                  "
    locate 13,20 : print "HIT ANY KEY TO CONTINUE"
    locate 23,1
    do
    loop while INKEY$ = ""
    goto 1

 61 cls
    color 0,7
    print "
    print "                    DISPLAY SELECTED INFORMATION FOR ALL POSITIONS
    print "
    locate 5,1  : print " 1 "
    locate 7,1  : print " 2 "
    locate 9,1  : print " 3 "
    locate 11,1 : print " 4 "
    locate 13,1 : print " 5 "
    locate 15,1 : print " 6 "
    locate 17,1 : print " 7 "
    locate 19,1 : print " 8 "
    locate 21,1 : print " 9 "
    locate 23,1 : print " 0 "
    color 7,0
    locate 5,6  : print "GENERAL DESCRIPTION"
    locate 7,6  : print "ELEMENT SPECIFICATION"
    locate 9,6  : print "ELEVATION ANGLE (DEG)"
    locate 11,6 : print "AZIMUTH ANGLE (DEG)"
    locate 13,6 : print "HEIGHT (CM)"
    locate 15,6 : print "RANGE (M)"
    locate 17,6 : print "EMISSIVITY"
    locate 19,6 : print "BACKGROUND DESCRIPTION"
    locate 21,6 : print "ELEMENT SPECIFICATION"
    locate 23,6 : print "POSITION CODE"
    locate 25,1 : print "CHOOSE FROM LIST, TYPE NUMBER";
 62 Y$     = INKEY$
    TITLE$ = "FOUT"
    for NB = 1 to NUMBER
      if Y$ = "1" then
        INFO$ (NB) = POST$ (NB)
        TITLE$     = " DESCRIPTION "
      elseif Y$ = "2" then
        INFO$ (NB) = ELPOST$ (NB)
        TITLE$     = "SPECIFICATION"
      elseif Y$ = "3" then
        INFO$ (NB) = STR$ (TETA (NB))
        TITLE$     = "  ELEVATION  "
      elseif Y$ = "4" then
        INFO$ (NB) = STR$ (AZIM (NB))
        TITLE$     = "  AZIMUTH    "
      elseif Y$ = "5" then
```

```
    INFO$ (NB) = STR$ (H (NB))
    TITLE$     = "  HEIGHT     "
  elseif Y$ = "6" then
    INFO$ (NB) = STR$ (RAN (NB))
    TITLE$     = "  RANGE      "
  elseif Y$ = "7" then
    INFO$ (NB) = STR$ (EPS (NB))
    TITLE$     = "  EMISSIVITY "
  elseif Y$ = "8" then
    INFO$ (NB) = SCREEN$ (NB)
    TITLE$     = " BACKGROUND  "
  elseif Y$ = "9" then
    INFO$ (NB) = ELSCR$ (NB)
    TITLE$     = "SPECIFICATION"
  elseif Y$ = "0" then
    INFO$ (NB) = POSCODE$ (NB)
    TITLE$     = "    CODE     "
  end if
  if TITLE$ = "FOUT" then
    NB = 0
    goto 62
  end if
next NB

cls
color 0,7
print "
print "                    DISPLAY SELECTED INFORMATION FOR ALL POSITIONS
print "
color 7,0
T$ = "POS "+TITLE$
locate 5,1 : print T$ tab(42) T$
for BN = 1 to NUMBER
  if BN < 16 then
    locate (6+BN),1                'LINES 7 TILL 17, COLUMNS  1 TILL 41
  else
    locate (BN-9),42               'LINES 7 TILL 17, COLUMNS 42 TILL 80
  end if
  print using "##" ; BN ; : print "   " ; INFO$(BN)
next BN
locate 25,1  : print "HIT ANY KEY TO CONTINUE" ;
locate 25,45 : print "F1 = RETURN TO MAIN MENU" ;
locate 23,1
do
loop while INKEY$ = ""
goto 32
```

Appendix B

Position file

```
 26
04-06-1990
GRASS                   CAMOUFLAGE-ADAPTIVE
GROWING-NORMAL          NO SUB-TITLE
-2001.00                +1236.00
-1323.00                -1428.00
   +0.00                   +0.00
 +340.00                  +90.00
   +6.00                   +1.00
  +10.00                   +8.00
   +0.95                   +0.67
GRASS                   CONCRETE
GROWING-NORMAL          NO SUB-TITLE
02518319025             33013216110
TREES-DECIDUOUS         CAMOUFLAGE-SCREEN
BUDS                    NO SUB-TITLE
-1996.00                 -566.00
 -896.00                -1051.00
  +90.00                   +0.00
 +340.00                 +200.00
 +300.00                   +2.00
  +30.00                  +25.00
   +0.95                   +0.95
TREES-DECIDUOUS         GRASS
BUDS                    CRUSHED
06428829064             32015229027
TREES-DECIDUOUS         CAMOUFLAGE-SCREEN
BUDS                    NO SUB-TITLE
-1538.00                 -555.00
 -930.00                -1067.00
  +90.00                   +0.00
 +290.00                 +200.00
 +200.00                   +2.00
  +40.00                  +25.00
   +0.95                   +0.91
GRASS                   GRASS
GROWING-NORMAL          CRUSHED
06427829025             32015229027
CONCRETE                TEST-SOURCE + 3 DEG
NO SUB-TITLE            NO SUB-TITLE
 +747.00                 -795.00
-1356.00                 -786.00
   +0.00                  +90.00
  +90.00                 +235.00
   +1.00                  +10.00
   +8.00                   +0.50
   +0.90                   +1.00
CONCRETE                SKY
NO SUB-TITLE            NO SUB-TITLE
11013219110             91026419440
```

Appendix C

CARABAS software listing

```
defdbl S
key off
color 0,7
locate 1,5   : print "
locate 2,5   : print "                    PHYSICS AND ELECTRONICS LABORATORY
locate 3,5   : print "
color 7,0
locate 5,5   : print "OUDE WAALSDORPERWEG 63  "
locate 6,5   : print "P.O. BOX 96864  "
locate 7,5   : print "2509 JG THE HAGUE  "
locate 8,5   : print "THE NETHERLANDS  "
locate 9,5   : print "TEL. (31) 70264221  "
locate 11,5  : print " CCCCCC     AAAAAA    RRRRRRR     AAAAAA    BBBBBBB
locate 12,5  : print "CCCCCCCC   AAAAAAAA   RRRRRRRR   AAAAAAAA   BBBBBBBB
locate 13,5  : print "CC         AA    AA   RR    RR   AA    AA   BB    BB
locate 14,5  : print "CC         AAAAAAAA   RRRRRRRR   AAAAAAAA   BBBBBBB
locate 15,5  : print "CC         AAAAAAAA   RRRRRRR    AAAAAAAA   BBBBBBB
locate 16,5  : print "CC         AA    AA   RR  RR     AA    AA   BB    BB
locate 17,5  : print "CCCCCCCC   AA    AA   RR   RR    AA    AA   BBBBBBBB
locate 18,5  : print " CCCCCC    AA    AA   RR    RR   AA    AA   BBBBBBB
locate 23,5  : print "(C) DEVELOPED BY Dr.ir. P.A.M. JACOBS AND Ing. R. van
locate 24,10 : print "LAST REVISED : 5 APRIL 1988";

for I=1 to 4000
next
'                    PROGRAMME NAME:  C A R A B A S

'                    PARAMETERS USED IN THIS PROGRAMME

'PARAMETERS RELATED TO THE DETECTOR WORKING IN :  3 - 5 µm     8 - 14 µm
'HAVE THE ARRAY INDEX EQUAL TO                  :    1           2
'TR(N,J,I)   :MEASURED RADIATION TEMPERATURE OF CALIBRATION SOURCE I
'             (I=1,3) AT LOCATION  X(N),Y(N)                              (K)
'TS(I..3)    :MEASURED TEMPERATURE OF CALIBRATION SOURCE  I               (K)
'E(J,I..3)   :CALCULATED 'APPARENT' PUPIL IRRADIANCE FOR CALIBRATION SOURCE
'             I (I=1,3) FOUND FROM THE CALIBRATION TABLE "CALIBRAT"    (W/m2)
'E(N,J,4)    :CALCULATED PUPIL IRRADIANCE FOR THE UNKNOWN SURFACE AT
'             LOCATION  X(N).Y(N)                                      (W/m2)
'V(N,J,I)    :MEASURED DETECTOR VOLTAGE FOR CALIBRATION SOURCE I  AT
'             LOCATION  X(N),Y(N)                                        (V)
'V(N,J,4)    :MEASURED DETECTOR VOLTAGE FOR THE UNKNOWN SURFACE  AT
'             LOCATION  X(N),Y(N)                                        (V)
'VPT(N,I)    :MEASURED VOLTAGES OVER THE Pt-RESISTORS IN THE CALIBRATION
'             SOURCES I (I=1,3) AT LOCATION X(N),Y(N)                    (V)
'A(J)        :CALIBRATION CONSTANT                                (W/(m2.V))
'B(J)        :CALIBRATION CONSTANT                                     (W/m2)
'            :THE CALIBRATION EQUATION IS GIVEN BY:
'             E(N,J,I) = A(N,J) * V(N,J,I) + B(N,J)                    (W/m2)
'ET(J,M)     :INTEGRATED PLANCK'S EQUATION FOUND FROM THE LOOK-UP TABLE
'             (SEE PROGRAMME "MAKETAB")
'TT(M)       :ARRAY OF TEMPERATURE DATA FOUND FROM THE LOOK-UP TABLE
'             THIS TABLE HAS A TEMPERATURE RESOLUTION OF 0.5 K
'             A LINEAR REGRESSION IS USED TO INCREASE RESOLUTION TO 0.1K
'X(N),Y(N)   :COORDINATES OF A MEASUREMENT LOCATION  N

'                    D E F I N E    K E Y F U N C T I O N S

'                    DISPLAY CURRENT DATA, DURING RUNNING OF PROGRAMME
on key (1) gosub 83
key (1) off

'                    DISPLAY THE TIMING PARAMETERS,
'                    DURING RUNNING OF THE PROGRAMME
on key (2) gosub 77
```

```
key (2) off

'                       INTERRUPT THE PROGRAMME, CLOSE THE FILES AND
'                       EXIT FROM THE PROGRAMME TO RETURN TO DOS
on key (3) gosub 36
key (3) off

'                       KEY 4 AND 5 TO PRINT RESULTS DURING RUNTIME
on key (4) gosub 49
key (4) on
on key (5) gosub 48
key (5) on

'                       KEY 6 TO CHANGE DATA DISKETTE
on key (6) gosub 79
key (6) off

'                       KEY 7 FOR EMERGENCY EXIT AT ALL TIME
on key (7) gosub 36
key (7) on

on ERROR goto 40

'                       MAXIMUM NUMBER OF CHANNELS = 30
const MAXDIM = 30
dim ET(2,200),TT(200)
dim CAL(3,9),RES(3),TS(3),CH(5),R(2),A(2),B(2)
dim VPT (MAXDIM,4),TIJD$ (MAXDIM),POST$ (MAXDIM)
dim V (MAXDIM,2,4),E (MAXDIM,2,4),TR (MAXDIM,2,4)
dim X (MAXDIM),Y (MAXDIM),TETA (MAXDIM),AZIM (MAXDIM)
dim H (MAXDIM),RAN (MAXDIM),EPS (MAXDIM)
dim ELPOST$ (MAXDIM),SCREEN$ (MAXDIM),ELSCR$ (MAXDIM)
dim POSCODE$ (MAXDIM)
dim MON$(12)
MON$(1)  = "JAN"
MON$(2)  = "FEB"
MON$(3)  = "MAR"
MON$(4)  = "APR"
MON$(5)  = "MAY"
MON$(6)  = "JUN"
MON$(7)  = "JUL"
MON$(8)  = "AUG"
MON$(9)  = "SEP"
MON$(10) = "OCT"
MON$(11) = "NOV"
MON$(12) = "DEC"
DEF FNDAG$(DAT$) = MID$(DAT$,4,3)+MON$(VAL(LEFT$(DAT$,2)))+RIGHT$(DAT$,5)

'                       O P E N   A   S E Q U E N C I A L   F I L E
'                       F O R   R E A D I N G   T H E   T A B L E S

'                       GET CALIBRATION TABLES FROM DISK UNDER THE NAME
'                       'CALIBRAT'
color 0,7
locate 6,39 : print "                                        "
locate 7,39 : print "     LOADING TABLES, PLEASE WAIT....    "
locate 8,39 : print "                                        "
color 7,0

'                       LOOK-UP TABLE FOR IRRADIANCE IS DEFINED FOR ABSOLUTE
'                       TEMPERATURES WITHIN THE RANGE 250 - 350 K.
open "I",#1,"CALIBRAT"
for M=1 to 200
  input #1,TT(M),ET(1,M),ET(2,M)
```

```
    next M

    '                    CALIBRATION TABLE FOR THE PLATINUM TEMPERATURE SENSOR
    '                    (Pt100), USABLE FOR MEASURED TEMPERATURES BETWEEN
    '                    -30 TO +50 degC.
    for I=1 to 9
      for J=1 to 3
        input #1,CAL(J,I)
      next J
    next I
    close #1

    '                    D E F A U L T    S E T T I N G S
    DRIVE$ = "D:"
    TEXT$  = ""
    DNEXT$ = ""
    TNEXT$ = ""
    PARSET = 0
    DIS    = 0
    ERDE   = 0
    INDEX  = 0
    FAC    = 0
    FOPEN  = 0
    TEL    = 0
    PR     = 0
    KEY4   = 0
    for J=1 to MAXDIM
      POST$(J) = ""
      TETA(J) = 0
      AZIM(J) = 0
      H(J) = 0
      RAN(J) = 1
      EPS(J) = 1
    next J
    out 1812,128          'AUTO INCREMENTING OFF

    '                    D E F I N E    M A I N - M E N U
1   open "I",#1,"A:POWER.FAL"
      input #1,POWER
    close #1
47  cls
    if POWER = 0 then 26
    TEST = 0
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "                    MAKE A CHOICE FROM THE MAIN MENU
    locate 3,5 : print "
    locate 5,5 : print " 1 "; : color 7,0  : print "   DEFINE PARAMETER SETTING
    locate 7,5 : color 0,7 : print " 2 "; : color 7,0 : print "   DISPLAY CURRE
    locate 9,5 : color 0,7 : print " 3 "; : color 7,0 : print "   PERFORM A TES
    locate 11,5: color 0,7 : print " 4 "; : color 7,0 : print "   START AUTOMAT
    locate 13,5: color 0,7 : print " e "; : color 7,0 : print "   EXIT"

2   A$ = INKEY$
    if A$ = "1" then
      goto 3
    elseif A$ = "2" then
      goto 29
    elseif A$ = "3" then
      cls
      TEST=1
      if PARSET = 1 then
        goto 34
      else
```

```
      goto 24
    end if
  elseif A$ = "4" then
    cls
    goto 34
  elseif A$ = "e" then
    cls
    close
    goto 35
  end if
  goto 2

  '                    D E F I N E   T I M I N G   P A R A M E T E R S
3 cls
  open "O",#1,"A:RESTUP.CAR"
4 color 0,7
  locate 1,5 : print "
  locate 2,5 : print "                          GIVE THE DATE TO START THE MEASUREME
  locate 3,5 : print "
  locate 5,5 : print " FORMAT  "; : color 7,0 : print " MM-DD-YYYY"
  locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " ACTUAL DA
  locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
  line input DSTART$
  if DSTART$ = "" then
    DSTART$ = DATE$
    goto 6
  elseif DSTART$ >= DATE$ then
    goto 6
  end if
  cls
  locate 2,2 : print "THE GIVEN DATE ";
  color 0,7 : print " " DSTART$ " ";
  color 7,0 : print " IS SMALLER THAN THE ACTUAL DATE ";
  color 0,7 : print " ";DATE$ " "
  color 7,0 : locate 4,2 : print "GIVE THE CORRECT ";
  goto 4

6 cls
  color 0,7
  locate 1,5 : print "
  locate 2,5 : print "                          GIVE THE DATE TO STOP THE MEASUREMEN
  locate 3,5 : print "
  locate 5,5 : color 0,7 : print " FORMAT  "; : color 7,0 : print " MM-DD-YYY
  locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " 12-31-199
  locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
  line input DSTOP$
  if DSTOP$ = "" then
    DSTOP$ = "12-31-1995"
    goto 21
  elseif DSTOP$ >= DATE$ then
    goto 21
  end if
  cls
  locate 2,2 : print "THE GIVEN DATE ";
  color 0,7 : print DSTOP$;
  color 7,0 : print " IS SMALLER THAN THE ACTUAL DATE ";
  color 0,7 : print " ";DATE$ " "
  color 7,0 : locate 4,2 : print "GIVE THE CORRECT ";
  goto 6

21 write #1,DSTOP$

22 cls
  color 0,7
```

```
    locate 1,5 : print "
    locate 2,5 : print "                    GIVE THE TIME TO START THE MEASUREME
    locate 3,5 : print "
    locate 5,5 : color 0,7 : print " FORMAT  "; : color 7,0 : print " HH:MM:SS
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " ACTUAL TI
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
    line input TSTART$
    if TSTART$ = "" then
      TSTART$ = "ACTUAL TIME"
      goto 23
    elseif TSTART$ >= TIME$ then
      goto 23
    end if
    cls
    locate 2,2 : print "THE GIVEN TIME ";
    color 0,7 : print " " TSTART$ " ";
    color 7,0 : print " IS SMALLER THAN THE ACTUAL TIME ";
    color 0,7 : print " ";TIME$" "
    color 7,0 : locate 4,2 : print "GIVE THE CORRECT ";
    goto 22

23  cls
    color 0,7
    locate 1,5 : print"
    locate 2,5 : print "                    GIVE THE TIME TO STOP THE MEASUREMEN
    locate 3,5 : print "
    locate 5,5 : print " FORMAT  "; : color 7,0 : print " HH:MM:SS "
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " 23:59:00"
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
    line input TSTOP$
    if TSTOP$ = "" then
      TSTOP$ = "23:59:00"
      goto 31
    elseif DSTOP$ > DSTART$ then
      goto 31
    elseif TSTOP$ > TIME$ then
      goto 31
    end if
    cls
    locate 2,2 : print "THE GIVEN TIME ";
    color 0,7 : print " " TSTOP$ " ";
    color 7,0 : print " IS SMALLER THAN THE ACTUAL TIME ";
    color 0,7 : print " ";TIME$" "
    color 7,0 : locate 4,2 : print "GIVE THE CORRECT ";
    goto 23

31  write #1,TSTOP$

    cls
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "                         GIVE THE TIME BETWEEN TWO SCANS
    locate 3,5 : print "
    locate 5,5 : print " FORMAT  "; : color 7,0 : print " MINUTES  "
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " 15 "
    locate 16,5: color 23,0: print " NOTE : "   : color 7,0
    locate 18,5: print "  A MULTIPLE OF INTERVALS SHOULD BE EXCACTLY ONE OUR (6
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : input " ",INTV
    if INTV < 1 then INTV = 15
    while 60 MOD INTV <> 0
      locate 10,5 : print "                                                  "
      locate 10,5 : color 0,7 : print "  INPUT  "; : color 7,0 : input " ",INTV
      if INTV < 1 then INTV = 15
    wend
```

```
    write #1,INTV

    PARSET = 1
 24 cls
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "  GIVE THE NAME OF THE FILE CONTAINING THE COORDINATE T
    locate 3,5 : print "
    DMY$ = FN DAG$(DATE$)
    PFIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "P"
    locate 5,5 : print " FORMAT  "; : color 7,0 : print " MMMDDYYP    "
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " "; PFIL$
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
    line input DESC$
    if DESC$ = "" then DESC$ = PFIL$

    write #1,DESC$
    write #1,DRIVE$
    close #1

    cls
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "                GIVE THE NAME OF THE DATA STORAGE FILE
    locate 3,5 : print "
    DMY$ = FN DAG$(DATE$)
    FIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "C"
    locate 5,5 : print " FORMAT  "; : color 7,0 : print " MMMDDYYC    "
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " "; FIL$
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
    line input FILENAME$
    if FILENAME$ = "" then FILENAME$ = FIL$

    cls
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "    ";
    color 7,0  : print "
    locate 2,76: color 0,7 : print "    "
    locate 3,1 : print "    ";
    color 7,0  : print "    INSERT DISKETTE WITH POSITION FILE  ";
    color 0,7  : print "  ";DESC$"  ";
    color 7,0  : print "  IN DRIVE D"
    locate 3,76: color 0,7 : print "    "
    locate 4,1 : print "    ";
    color 7,0  : print "
    locate 4,76: color 0,7 : print "    "
    locate 5,1 : print "    ";
    color 7,0  : print "    LEAVE THE QB-SYSTEM DISKETTE IN DRIVE A:       "
    locate 5,76: color 0,7 : print "    "
    locate 6,1 : print "    ";
    color 7,0  : print "
    locate 6,76: color 0,7 : print "    "
    locate 7,1 : print "    ";
    color 7,0  : print "
    locate 7,1 : color 0,7 : print "
    color 7,0
    locate 12,1: print "HIT ANY KEY WHEN YOU ARE READY"
    while V$ = ""
      V$ = INKEY$
    wend
    if POWER = 1 then 28
 26 open "I",#1,"A:RESTUP.CAR"
```

```
      input #1,DSTOP$
      input #1,TSTOP$
      input #1,INTV
      input #1,DESC$
      input #1,DRIVE$
    close #1
    DMY$ = FN DAG$(DATE$)
    FIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "C"
    FILENAME$ = FIL$
    gosub 27
28  cls
    color 0,7
    locate 10,20 : print "                                    "
    locate 11,20 : print "    PREPARING DISK FOR WRITING      "
    locate 12,20 : print "                                    "
    color 7,0

    '                     OPEN DUMMY FILE WHICH WILL BE KILLED LATER
    '                     TO BE ABLE TO CLOSE FILES IF DISK IS FULL ERROR OCCURS
    close #3
    open "O",#3,DRIVE$+"DUMMY.CAR"

      '                   WRITE 2401 BYTES TO THIS FILE
      for JJ = 1 to 100
        write #3,"BBBBBBBBBBBBBBBBBBBB"
      next JJ
    close #3

    '                     OPEN ERROR COMMUNICATION FILE ON #3
    open DRIVE$+"ERROR.CAR" for append as #3
    fopen = 1

    '                     OPEN A SEQUENCIAL FILE, NAMED "DESC$" ON CHANNEL 1
    '                     TO READ POSITIONS
    open "I",#1,DRIVE$+DESC$
      input #1 , NUMBER
      input #1 , DAY$
      for N = 1 to NUMBER
        input #1 , POST$(N)
        input #1 , ELPOST$(N)
        input #1 , X(N)
        input #1 , Y(N)
        input #1 , TETA(N)
        input #1 , AZIM(N)
        input #1 , H(N)
        input #1 , RAN(N)
        input #1 , EPS(N)
        input #1 , SCREEN$(N)
        input #1 , ELSCR$(N)
        input #1 , POSCODE$(N)
      next N
    close #1

    '                     OPEN A SEQUENCIAL DATA OUTPUTFILE, NAMED "FILENAME$"
    '                     ON CHANNEL 2
    close #2
    open DRIVE$+FILENAME$ for append as #2
    if POWER = 0 then 42
    if TEST = 1 then 34 else 47

    '                     DISPLAY CURRENT PARAMETERS
29  if INTV >= 1 then 32
    cls
    color 0,7
```

```
    locate 9,20 : print "                                          "
    locate 10,20: print "                                          "
    locate 11,20: print "   TIMING PARAMETERS NOT DEFINED YET      "
    locate 12,20: print "                                          "
    locate 13,20: print "                                          "
    locate 20,20 : color 7,0 : print "HIT ANY KEY WHEN YOU ARE READY"
    while V$ = ""
      V$ = INKEY$
    wend
    goto 47

    LENG = LEN(TSTART$)
    if LENG >= 8 then 32
    for I = 1 to 8-LENG
      TSTART$ = TSTART$ + "0"
    next I
    MID$(TSTART$,3,1) = ":"
    MID$(TSTART$,6,1) = ":"

 32 cls
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "                    CURRENT TIMING PARAMETERS SETTINGS
    locate 3,5 : print "
    locate 6,5 : print "START DATE" TAB(30) ":" ; DSTART$ : print
    locate 8,5 : print "STOP DATE" TAB(30) ":" ; DSTOP$ : print
    locate 10,5: print "START TIME" TAB(30) ":" ; TSTART$ : print
    locate 12,5: print "STOP TIME" TAB(30) ":" ; TSTOP$ : print
    locate 14,5: print "SCAN INTERVAL" TAB(30) ":" ; INTV " (min)"
    locate 16,5: print "COORDINATE FILENAME" TAB(30) ":" ; DESC$
    locate 18,5: print "DATA STORAGE FILENAME" TAB(30) ":"; FILENAME$

    locate 24,5: print "HIT ANY KEY TO CONTINUE";
    while V$ = ""
      V$ = INKEY$
    wend
    goto 47

    '                   S T A R T   O F   M E A S U R E M E N T S

    '                   CHECK IF THE REMOTE CONTROL IS DISCONNECTED,
    '                   BY READING THE MIRROR POSITION TWICE.
    '                   IF NOT CHANGED AFTER 1 SECOND, MIRROR IS STATIC.
 34 ROT0 = INP(1820)
    for TELL = 1 to 3
      PP$ = TIME$
      while TIME$ <= PP$
      wend
    next TELL
    if (INP(1820)-ROT0) = 0 then 37

    cls
    color 0,7
    locate 10,20 : print "                                  "
    locate 11,20 : print "  DISCONNECT THE REMOTE CONTROL   "
    locate 12,20 : print "                                  "
    color 7,0
    locate 16,20 : print "HIT ANY KEY WHEN READY"
    while V$ = ""
      V$ = INKEY$
    wend

    '                   CHECK IF A TESTRUN SHOULD BE PERFORMED
 37 if TEST = 1 then
```

```
      TSTART$ = TIME$
      goto 46
    end if
    '                        CHECK IF TIMING PARAMETERS ARE SET
    if PARSET = 1 then 41
 39 cls
    color 0,7
    locate 8,20 : print "                                          "
    locate 9,20 : print "    DEFINE TIMING PARAMETERS FIRST        "
    locate 10,20: print "                                          "
    color 7,0
    locate 15,20: print "HIT ANY KEY TO CONTINUE"
    while V$ = ""
      V$ = INKEY$
    wend
    goto 47

    '                        CHECK FOR IMMEDIATE START OF THE MEASUREMENT CYCLE
 41 if DSTART$ = DATE$ then
      if TSTART$ = "ACTUAL TIME" then
        TSTART$ = TIME$
        goto 45
      end if
    end if

    '                        WAIT FOR START MOMENT
 42 LENG = LEN(TSTART$)
    if LENG < 8 then
      for I = 1 to 8-LENG
        TSTART$ = TSTART$ + "0"
      next I
      MID$(TSTART$,3,1) = ":"
      MID$(TSTART$,6,1) = ":"
    end if

    cls
    color 0,7
    locate 1,1  : print "
    locate 2,1  : print "                      WAITING FOR THE BEGIN OF THE MEASUR
    locate 3,1  : print "
    locate 6,17 : print "                    " : locate 6,47 : print "
    locate 7,17 : print "   ACTUAL DATA      " : locate 7,47 : print "   START DAT
    locate 8,17 : print "                    " : locate 8,47 : print "
    locate 16,17: print "                                            "
    locate 17,17: print "          Data will be stored on file       "
    locate 18,17: print "                                            "
    locate 19,17: print "                  " ; DRIVE$+FIL$ ; "
    locate 20,17: print "                                            "
    color 7,0
    locate 10,50: print DSTART$
    locate 12,51: print TSTART$
 44 locate 10,20: print DATE$
    locate 12,21: print TIME$
    if DATE$ = DSTART$ then
      if TIME$ = TSTART$ then
        goto 45
      elseif TIME$ > TSTART$ then
        gosub 27
        locate 12,51: print TSTART$
      end if
    elseif DATE$ > DSTART$ then
      gosub 27
      locate 10,50: print DSTART$
    end if
```

```
      if RIGHT$(TIME$,2) = "25" then
        gosub 25
      end if
      goto 44

   45 DNEXT$ = DSTART$
      TNEXT$ = TSTART$

   46 if PR = 0 then
        cls
        print "BEGIN OF CYCLE  "; TIME$
        TNUL$ = TIME$
      end if
      for N = 1 to NUMBER
        locate 25,1 : print "
        locate 25,1 : print" READY ..............  GOING FOR POSITION ";

        '                   START OF MEASUREMENT CYCLE
        locate 25,41 : print N;

        '                   START SCANNING MIRROR, OPEN THE HATCH AND UNLOCK
        PORTB = 2
        gosub 76

        '                   WAIT 3 SECONDS TO OPEN THE HATCH COMPLETELY.
        if N <= 1 then
          TELL = 0
          PP$ = TIME$
          while TELL < 4
            TELL = TELL + 1
            while TIME$ <= PP$
            wend
            PP$ = TIME$
          wend
        end if

        '                   BRING THE SENSOR HEAD TO POSITION N
        DAC = 0
        VX = X(N)
        gosub 85
        DAC = 1
        VY = Y(N)
        gosub 84
        gosub 86

        '                   SENSOR HEAD REACHED POSITION N
        locate 25,1  : print "REACHED POSITION ";
        locate 25,18 : print N ".............  SCANNING NOW .......                "

        '                   LOCK THIS POSITION
        PORTB = 0
        gosub 76

        '                   CONTROL REGISTER 1820 IS USED AS:
        'BITS *  7  6 *       5       *   4   *       3        *  2    1  *   0
        '     *  FREE * POWER FAILURE * RAIN * POWER FAILURE *  MIRROR  * MIRROR
        '     *       *      MAINS     * GAUGE*     CARABAS    * POSITION * TRIGGER

        TIJD$(N) = TIME$
        for COUNT = 1 to 4

          '                 WHEN ENTERING THIS LOOP, WAIT FOR MIRROR POSITION 4
          '                 THAT IS WHEN THE 8-12 UM DETECTOR LOOKS TO THE OUTSIDE
          '                 WORLD FOR CLAMPING THE 3-5 UM DETECTOR ON THE REARSIDE
```

```
        '                SOURCE.
        '                DETECT POSITIVE EDGE OF 'MIRROR IN POSITION' PULS
50      wait 1820,1,1
        wait 1820,1,0

        MIRROR = INT ((INP(1820) and 7) /2 + 1) + 2
        if MIRROR > 4 then
          MIRROR = MIRROR - 4
        end if
        if COUNT >= 2 then
          goto 52
        elseif MIRROR <> 4 then
          goto 50
        end if

        '                READ VOLTAGE OVER Pt RESISTORS
52      out 1813,4
        out 1814,0
        wait 1812,128,0                          ' WAIT FOR CONVERSION READY
        VPT(N,MIRROR) = INP(1813) + 256*INP(1814)

        '                DETECT NEGATIVE EDGE OF 'MIRROR IN POSITION' PULS
        wait 1820,1,0
        wait 1820,1,1

        '                READ INPUT VALUE FOR BOTH DETECTORS:
        '                DAC = 2          DETECTOR SIGNAL  3 - 5  MICRON
        '                DAC = 3          DETECTOR SIGNAL  8 - 14  MICRON
        out 1813,2
        out 1814,0
        wait 1812,128,0                          ' WAIT FOR CONVERSION READY
        V(N,1,MIRROR) = INP(1813) + 256*INP(1814)
        out 1813,3
        out 1814,0
        wait 1812,128,0                          ' WAIT FOR CONVERSION READY
        V(N,2,MIRROR) = INP(1813) + 256*INP(1814)
      next COUNT
      for K = 1 to 4
        if VPT(N,K) > 32767 then VPT(N,K) = VPT(N,K) - 65536!
        if V(N,1,K) > 32767 then V(N,1,K) = V(N,1,K) - 65536!
        if V(N,2,K) > 32767 then V(N,2,K) = V(N,2,K) - 65536!
      next K
    next N

    '                RETURN TO THE FIRST POSITION,
    '                STOP THE MIRROR AND CLOSE THE HATCH

    locate 25,1
    print "STOP MIRROR, CLOSE HATCH AND RETURN TO FIRST POSITION..... ";TIME$
    PORTB = 3
    gosub 76
    DAC = 0
    VX = X(1)
    gosub 85
    DAC = 1
    VY = Y(1)
    gosub 84

    '                WAIT HERE UNTIL SENSOR HAS REACHED THE STARTING
    '                POSITION AND LOCK IT.
    gosub 86

    '                E N D    O F    M E A S U R M E N T    C Y C L E
```

```
'                    S T A R T    C O N V E R S I O N      O F      D A T A

'                    DETERMINE CALIBRATION CONSTANTS A(J) AND B(J).
'                    MEASURED VOLTAGES OVER THE Pt100 SENSORS  VPT(N,I)
'                    MUST BE KNOWN AT THIS POINT, AS WELL AS THE DETECTOR
'                    VOLTAGES V(N,J,I) FOR THE CALIBRATION SOURCES.

locate 25,1
print " CALCULATING TEMPERATURES NOW ..... ";TIME$
for N = 1 to NUMBER
  'for J = 1 to 2
    J = 2                                          ' ONLY 10µ DETECTOR IN USE
    if J = 2 then 54

    '                PUT THE DATA IN THE CORRECT ORDER FOR DETECTOR J=1

    HELP1 = V(N,J,3)
    HELP2 = V(N,J,4)
    HELP3 = V(N,J,1)
    HELP4 = V(N,J,2)
    V(N,J,1) = HELP1
    V(N,J,2) = HELP2
    V(N,J,3) = HELP3
    V(N,J,4) = HELP4

    '                DETERMINE THE CALIBRATION CONSTANTS A(J) AND B(J)
54  gosub 15
    E(N,J,4) = B(J)*V(N,J,4) + A(J)

    '                USE LOOK-UP TABLE TO FIND THE CORRESPONDING
    '                TEMPERATURES.
    for M = 1 to 199
      if E(N,J,4) >= ET(J,M) and E(N,J,4) <= ET(J,M+1) then
        TR(N,J,4) = TT(M)
        goto 56
      elseif M < 199 then
        goto 55
      end if

      '              E(N,J,4) IS OUTSIDE THE TABLE-RANGE:
      '              WRITE TR(N,J,4) = 0 TO OUTPUT
      TR(N,J,4) = 0
      goto 20
55  next M

    '                THE TEMPERATURE FOUND TR(N,J,4) LIES BETWEEN
    '                TWO NODES TT(M) SEPARATED BY 0.5 K.
    '                ASSUMING A LINEAR DEPENDENCE OF THE IRRADIANCE
    '                WITH RESPECT TO TEMPERATURE, IN BETWEEN THE 2 NODES,
    '                A MORE ACCURATE VALUE CAN BE OBTAINED BY LINEAR
    '                INTERPOLATION.

56  TR(N,J,4) = TT(M)+(TT(M+1)-TT(M))*(E(N,J,4)-ET(J,M))/(ET(J,M+1)-ET(J,M)
    TR(N,J,4) = TR(N,J,4) - 273
20  'NEXT J              NEEDED IF 2 DETECTORS ARE IN USE

    if DIS = 0 then 60

    '                PRINT RESULTS DURING RUNTIME

    lprint "RECORDING TIME" TAB(25) ":"; TIJD$(N)
    lprint "MEASUREMENT POINT" TAB(25) ":"; N
    lprint : lprint  "SOURCE" TAB(30) "1" TAB(45) "2" TAB(60) "3"
```

```
      lprint "-----------------------------------------------------------------
      lprint TAB(3) "Pt - RESISTANCE" TAB(28) RES(1) TAB(43) RES(2) TAB(58) RES
      lprint TAB(3) "Pt - VOLTAGE" TAB(28) VPT(N,1);TAB(43) VPT(N,2);TAB(58) VP
      lprint TAB(3) "Pt-TEMPERATURE" TAB(28);TS(1)-273 TAB(43);TS(2)-273 TAB(58
      lprint : lprint TAB(3) "DETECTOR VOLTAGE 3-5" TAB(28);V(N,1,1) TAB(43);V(
      lprint TAB(3) "ENERGY IN  3-5" TAB(28);E(N,1,1) TAB(43);E(N,1,2) TAB(58);
      lprint : lprint TAB(3) "DETECTOR VOLTAGE 8-14" TAB(28);V(N,2,1) TAB(43);V
      lprint TAB(3) "ENERGY IN  8-14" TAB(28);E(N,2,1) TAB(43);E(N,2,2) TAB(58)
      lprint : lprint "CALIBRATION PARAMETERS" TAB(32) "3-5 UM" TAB(62) "8-14 U
      lprint "-----------------------------------------------------------------
      lprint TAB(3) "CONSTANT B" TAB(30) B(1) TAB(60) B(2)
      lprint TAB(3) "CONSTANT A" TAB(30) A(1) TAB(60) A(2)
      lprint TAB(3) "CORRELATION COEFFICIENT" TAB(30) R(1) TAB(60) R(2)
      lprint
      lprint "DETECTOR VOLTAGE" TAB(30) V(N,1,4) TAB(60) V(N,2,4)
      lprint "MEASURED TEMPERATURE" TAB(30) TR(N,1,4) TAB(60) TR(N,2,4)
      lprint
      lprint
      lprint

60 next N

   '                  C O N V E R S I O N    C O M P L E T E D

   '                  W R I T E   D A T A   T O   D I S K

   '                  IF A TESTRUN WAS PERFORMED, THE PROGRAMME
   '                  RETURNS TO THE MAIN MENU.
   '                  DURING AUTOMATED OPERATION THE TIMING PARAMETERS
   '                  DETERMINE THE TIME BEHAVIOUR.

   '                  WRITE DATA TO THE FILE "FILENAME$" THROUGH CHANNEL #2
61 print #2,DESC$
   print #2,USING "###";NUMBER
   print #2,USING "\        \";DNEXT$
   for N = 1 to NUMBER
     print #2,USING "\      \";TIJD$(N)
     'When 5µ detector is in use :print #2,USING "+####.##";TR(N,1,4)
     print #2,USING "+####.##";TR(N,2,4)
   next N

   if TEST = 0 then 63
   DAC = 0
   VX = X(1)
   gosub 85
   DAC = 1
   VY = Y(1)
   gosub 84
   gosub 86
   locate 3,1 : print "END OF CYCLE    "; TIME$
   TST$ = TIME$
   AA = 0
   IF VAL(MID$(TST$,7,2)) >= VAL(MID$(TNUL$,7,2)) then
     AA=1
   end if
   DIF = (VAL(MID$(TST$,1,2))-VAL(MID$(TNUL$,1,2)))*60 + (VAL(MID$(TST$,4,2))-
   locate 5,1 : print "MINIMUM SCAN INTERVAL ";
   color 0,7  : print " ";DIF;
   color 7,0  : print " MINUTES"
   locate 25,1: print "HIT ANY KEY TO RETURN TO THE MAIN MENU";
   while V$ = ""
     V$=INKEY$
   wend
   goto 47
```

```
'                    C H E C K   F O R   S T O P   T I M E   A N D   D A T E

63 if DATE$ >= DSTOP$ and TIME$ >= TSTOP$ then 36

'                    CALCULATION OF NEXT START TIME :TNEXT$
   TOUD$ = TNEXT$
   M = VAL(MID$(TNEXT$,4,2))
   H = VAL(MID$(TNEXT$,1,2))
   M = M + INTV
   while M >= 60
     H = H + 1
     M = M - 60
   wend
   if H >= 24 then
     FAC = 1
     H = H - 24
   end if
   if H < 10 then
     H$ = "0" + MID$(STR$(H),2,2)
     goto 66
   else
     H$ = MID$(STR$(H),2,2)
   end if
66 if M < 10 then
     M$ = "0" + MID$(STR$(M),2,2)
     goto 67
   else
     M$ = MID$(STR$(M),2,2)
   end if
67 TNEXT$ = H$ + ":" + M$ + ":00"

'                    DETERMINE NEXT START DAY IF OLD ONE IS EXPIRED
'                    (I.E. "FAC=1")
   if FAC=0 then 75
   FAC = 0
   DAY = VAL(MID$(DNEXT$,4,2))
   MONTH = VAL(MID$(DNEXT$,1,2))
   YEAR = VAL(MID$(DNEXT$,7,4))
   NDAY = DAY + 1
   on MONTH goto 68,69,68,70,68,70,68,68,70,68,70,68
68 DAY = 31
   goto 71
69 DAY = 28
   X = YEAR MOD 4
   IF X = 0 then DAY = 29
   goto 71
70 DAY = 30
71 if NDAY <= DAY then
     goto 72
   else
     NDAY = NDAY - DAY
     MONTH = MONTH + 1
   end if
   if MONTH <= 12 then
     goto 72
   else
     MONTH = 1
     YEAR = YEAR + 1
   end if
72 if NDAY < 10 then
     DAYST$ = "0" + MID$(STR$(NDAY),2,2)
     goto 73
   else
```

```
      DAYST$ = MID$(STR$(NDAY),2,2)
    end if
73  if MONTH < 10 then
      MONTH$ = "0" + MID$(STR$(MONTH),2,2)
      goto 74
    else
      MONTH$ = MID$(STR$(MONTH),2,2)
    end if
74  YEAR$ = MID$(STR$(YEAR),2,4)
    DNEXT$ = MONTH$ + "-" + DAYST$ + "-" + YEAR$
87  DMY$ = FN DAG$(DNEXT$)
    close #2
    FIL$ = MID$(DMY$,4,3) + LEFT$(DMY$,2) + RIGHT$(DMY$,2) + "C"
    FILENAME$ = FIL$
    open DRIVE$+FILENAME$ for append as #2

    if TNEXT$ = "00:00:00" and DATE$ < DNEXT$ then
      goto 88
    end if
75  if TIME$ > TNEXT$ or DATE$ > DNEXT$ then
      gosub 27
      TNEXT$ = TSTART$
      DNEXT$ = DSTART$
      if TNEXT$ = "00:00:00" then
        goto 87
      end if
    end if

    '                    DISPLAY SELECTED DATA CHANNELS
    '                    PRINT RESULT WITHOUT REFRESHING THE SCREEN
88  if PR = 1 then 19

    '                    DISPLAY TIMING PARAMETERS
77  PR = 0
    cls
    color 7,0
    locate 25,1: print "F1=DISPLAY DATA CHANNELS       F6=CHANGE DATA DISKETTE
    color 0,7
    locate 1,1 : print "
    locate 2,1 : print "                              WAITING FOR BEGIN OF NEXT SCAN
    locate 3,1 : print "
    locate 6,3 : print "                        " : locate 6,33 : print "
    locate 7,3 : print "   ACTUAL DATA          " : locate 7,33 : print "    NEXT SCAN
    locate 8,3 : print "                        " : locate 8,33 : print "
    BEGIN$ = DRIVE$ + FIL$
    locate 16,1: print "
    locate 17,1: print "                               Data will be stored on file
    locate 18,1: print "
    locate 19,1: print "                                    "; BEGIN$ ;"
    locate 20,1: print "
78  color 7,0
    locate 10,6 : print DATE$
    locate 10,36: print DNEXT$
    locate 10,64: print DSTOP$
    locate 12,7 : print TIME$
    locate 12,37: print TNEXT$
    locate 12,65: print TSTOP$

    '                    THE PARAMETER "FAC" CONTROLES THE CHANGE OF DATE,
    '                    SO THAT THE CHECK FOR THE NEXT START TIME CAN BE DONE
    '                    CORRECTLY. CHECK FOR MAIN POWER FAILURE WHILE
    '                    WAITING FOR THE NEXT MEASUREMENT CYCLE
    key (1) on
    key (2) on
```

```
    key (3) on
    key (6) on
    if TIME$ >= TNEXT$ and DATE$ >= DNEXT$ then
      key (1) off
      key (2) off
      key (3) off
      key (6) off
      goto 46
    else
      gosub 25
      goto 78
    end if

    '                    C H A N G E    D A T A    D I S K E T T E
 79 close
    kill DRIVE$+"DUMMY.CAR"
    if ERDE = 0 then kill DRIVE$+"ERROR.CAR"
    cls
    color 0,7
    locate 9,20 : print "                                                    "
    locate 10,20: print "                                                    "
    locate 11,20: print "  PUT AN EMPTY DATA DISKETTE IN ";
    color 7,0 : print "   DRIVE D     ";
    color 0,7 : print " "
    locate 12,20: print "                                                    "
    locate 13,20: print "                                                    "
    color 7,0
    locate 20,20: print "HIT ANY KEY WHEN YOU ARE READY"
    while V$ = ""
      V$ = INKEY$
    wend
    DRIVE$ = "D:"
    open "O",#1,"A:RESTUP.CAR"
      write #1,DSTOP$
      write #1,TSTOP$
      write #1,INTV
      write #1,DESC$
      write #1,DRIVE$
    close #1
    cls
    color 0,7
    locate 2,1 : print "                                                        "
    locate 3,1 : print "   DO YOU WANT TO CHANGE THE DATA FILENAME   (Y/N)    "
    locate 4,1 : print "                                                        "
    color 7,0
 8  V$ = INKEY$
    if V$ = "n" OR V$ = "N" then
      goto 82
    elseif V$ <> "y" and V$ <> "Y" then
      goto 81
    end if
    cls
    color 0,7
    locate 1,5 : print "
    locate 2,5 : print "                              GIVE THE DATA FILENAME
    locate 3,5 : print "
    HILFE$ = FILENAME$
    locate 5,5 : print " FORMAT  "; : color 7,0 : print " MMMDDYYC    "
    locate 7,5 : color 0,7 : print " DEFAULT "; : color 7,0 : print " LAST GIVE
    locate 10,5: color 0,7 : print "  INPUT  "; : color 7,0 : print " ";
    line input FILENAME$
    if FILENAME$ = "" then FILENAME$ = HILFE$

 82 cls
```

```
    locate 10,20: color 0,7 : print "                                    "
    locate 11,20: print "    PREPARING DISK FOR WRITING    "
    locate 12,20: print "                                  "
    color 7,0
    open "D:"+FILENAME$ for append as #2

    '                    COPY THE POSITION FILE TO THE NEW DATA DISKETTE

    open "O",#1,"D:"+DESC$
    print #1 , USING "###" ; NUMBER
    print #1 , USING "\        \" ; DAY$
    for N = 1 to NUMBER
      print #1 , USING "\                          \" ; POST$(N)
      print #1 , USING "\                        \" ; ELPOST$(N)
      print #1 , USING "+####.##" ; X(N)
      print #1 , USING "+####.##" ; Y(N)
      print #1 , USING "+####.##" ; TETA(N)
      print #1 , USING "+####.##" ; AZIM(N)
      print #1 , USING "+####.##" ; H(N)
      print #1 , USING "+####.##" ; RAN(N)
      print #1 , USING "+####.##" ; EPS(N)
      print #1 , USING "\                          \" ; SCREEN$(N)
      print #1 , USING "\                        \" ; ELSCR$(N)
      print #1 , USING "\                \" ; POSCODE$(N)
    next N

    '                    OPEN DUMMY FILE WHICH WILL BE KILLED LATER,
    '                    TO BE ABLE TO CLOSE FILES IF DISK FULL ERROR OCCURS
    close #3
    open "O",#3,"D:DUMMY.CAR"

      '                  WRITE 2401 BYTES TO THIS FILE
      for JJ = 1 to 100
        write #3,"DDDDDDDDDDDDDDDDDDDD"
      next JJ
    close #3

    '                    OPEN ERROR COMMUNICATION FILE ON #3
    open "D:ERROR.CAR" for append as #3
    fopen = 1
    return 77

    '                    S U B R O U T I N E S

    '                    DISPLAY DATA DURING RUNNING OF THE PROGRAMME
    '                    AFTER A SCAN IS COMPLETED AND THE CONVERSION IS
    '                    TERMINATED, A MAXIMUM NUMBER OF 4 POSITIONS CAN BE
    '                    DISPLAYED ON SCREEN.
 83 cls
    TEL = 0
    PR = 1
    key (1) off
 51 line input "GIVE POSITIONS SEPARATED BY A COMMA (4 MAXIMUM) :"; POSI$
    ASTART = 1
    HALT = 0
    CHAN = 1
 53 COMMA = INSTR( ASTART , POSI$ , "," )
    if COMMA = 0 then
      COMMA = INSTR( ASTART , POSI$ , "" )
      HALT = 1
      LENGTE = COMMA + 1
      goto 58
    end if
    LENGTE = COMMA - 1
```

```
 58 CH(CHAN) = VAL(MID$( POSI$ , ASTART , LENGTE ))
    if CH(CHAN) < 1 or CH(CHAN) > NUMBER then
      cls
      print "POSITION ";CH(CHAN) "HAS NOT BEEN DEFINED, TRY AGAIN"
      print
      print
      print
      goto 51
    end if
    if HALT = 1 then
      goto 18
    else
      ASTART = LENGTE + 2
      CHAN = CHAN + 1
      if CHAN > 4 then
        CHAN = 4
        goto 18
      else
        goto 53
      end if
    end if
 19 if TEL > 0 and TEL < 15 then
      goto 17
    end if
 18 cls                    'REGULATES SCREEN REFRESH
    locate 2,1
    TEL = 0
    print "--------------------------------------------------------------------------------
    locate 3,25 : print "DATE      "; : print DATE$
    print "--------------------------------------------------------------------------------
    locate 5,11 : print "|"
    for J = 1 to CHAN
      locate 5,J*17-5 : print "   POSITION "; : print CH(J);
      if CH(J) < 10 then
        print  SPC(1)
      end if
      print "|"
    next J
    locate 6,1 : print "----------|"
    for J = 1 to CHAN
      locate 6,J*17-5 : print "----------------|"
    next J
    locate 7,1 : print "TIME"
    locate 7,11: print "|"
    for J = 1 to CHAN
      locate 7,J*17-5 : print "  3-5" SPC(5) "8-14  |"
    next J
    locate 8,1 : print "----------|"
    for J = 1 to CHAN
      locate 8,J*17-5 : print "----------------|"
    next J
    locate 9,1
 17 locate 9+TEL,1 : print MID$(TOUD$,1,5)        ' JUST PRINT HOUR AND MINUTES
    locate 9+TEL,11: print "|"
    for J = 1 to CHAN
      locate 9+TEL,J*17-5 : print USING "###.##";TR(CH(J),1,4);
      print SPC(3);
      print USING "###.##";TR(CH(J),2,4)
      locate 9+TEL,J*17+11: print "|"
    next J
    TEL = TEL + 1
    locate 25,1 : print "
 16 locate 25,1 : color 0,7 : print " ACTUAL TIME ";
    color 7,0   : print " ";TIME$;"          ";
```

```
      color 0,7    : print " NEXT SCAN ";
      color 7,0    : print " ";TNEXT$;
      locate 25,59: print "F2=TIMING PARAMETERS";
      key (1) on
      key (2) on
      if right$ (TIME$,2) = "25" then
        gosub 25
      end if
      if TIME$ >= TNEXT$ and DATE$ = DNEXT$ then
        key (1) off
        key (2) off
        goto 46
      else
        key (1) stop
        key (2) stop
        goto 16
      end if
      return

      '                    C A L I B R A T I O N

      '                    FIRST THE MEASURED VOLTAGE VPT(N,I) OVER THE Pt100
      '                    SENSORS IS CONVERTED TO RESISTANCE VALUES 'RES(I)'
      '                    DEPENDING ON THE IMPEDANCES USED IN THE ELECTRONIC
      '                    CIRCUITS. THIS IS A LINEAR OPERATION THROUGH A
      '                    REGRESSION CONSTANT: 'ALPHA'
   15 ALPHA = .013441
      OFFSET = 100.087
      for I = 1 to 3
        RES(I) = ALPHA * VPT(N,I) + OFFSET
      next

      '                    NEXT DETERMINE IN WHICH INTERVAL THE RESISTANCE
      '                    RES(I) OF CALIBRATION SOURCE I IS FOUND.
      '                    K1 GIVES THE LOWER BOUNDARY.
      for I = 1 to 3
        K1 = 1
        for K = 1 to 8
          if RES(I) >= CAL(1,K)  then
            K1 = K
          end if
        next K

        '                  NEXT DETERMINE THE TEMPERATURE OF CALIBRATION
        '                  SOURCE I, KNOWING ITS RESISTIVE VALUE  RES(I), FROM:
        TS(I) = 273! + CAL(3,K1) + (RES(I) - CAL(1,K1)) / CAL(2,K1)          ' (K)
      next I

      '                    NEXT USE THE LOOK-UP TABLE TO DETERMINE THE
      '                    'APPARENT' PUPIL IRRADIANCE FOR CALIBRATION SOURCE I.
      for L = 1 to 3
        for M = 1 to 199
          if TS(L) >= TT(M) and TS(L) <= TT(M+1) then
            E(M,J,L) = ET(J,M)
            goto 12
          elseif M < 199 then
            goto 13
          else
            goto 14
          end if

          '                CALIBRATION NOT POSSIBLE BECAUSE TS(L) IS OUT OF
          '                THE TABLE-RANGE
   14     A(J) = 0
```

```
        B(J) = 0
        return
13    next M

      '                    THE IRRADIANCE FOUND E(N,J,I) LIES BETWEEN TWO NODES
      '                    ET(J,M) SEPARATED BY 0.5 K. ASSUMING A LINEAR
      '                    DEPENDENCE OF THE IRRADIANCE WITH RESPECT TO
      '                    TEMPERATURE, IN BETWEEN THE 2 NODES, A MORE ACCURATE
      '                    VALUE CAN BE OBTAINED BY LINEAR INTERPOLATION.

12    E(N,J,L) = ET(J,M) + (ET(J,M+1)-ET(J,M)) * (TS(L)-TT(M)) / (TT(M+1)-TT(M)
    next L

      '                    CALCULATION OF THE CALIBRATION CONSTANTS A(J) AND
      '                    B(J) USING THREE CALIBRATION SOURCES. A(J) AND B(J)
      '                    ARE THE RESULT OF A LINEAR REGRESSION FIT OF PAIRS
      '                    V(N,J,I) AND E(N,J,I).R IS THE CORRELATIONCOEFFICIENT.
    SOMV = 0
    SOMV2 = 0
    SOME = 0
    SOMEV = 0
    SOME2 = 0
    for I = 1 to 3
      SOMV = SOMV + V(N,J,I)
      SOMV2 = SOMV2 + V(N,J,I)^2
      SOME = SOME + E(N,J,I)
      SOME2 = SOME2 + E(N,J,I)^2
      SOMEV = SOMEV + E(N,J,I)*V(N,J,I)
    next I

    B(J) = (SOME*SOMV-3*SOMEV) / (SOMV*SOMV-3*SOMV2)
    A(J) = (SOME-B(J)*SOMV) / 3
    R(J) = (A(J)*SOME+B(J)*SOMEV-(SOME^2)/3) / (SOME2-(SOME^2)/3)

      '                    NEXT THE MEASURED VOLTAGES OF THE UNKNOWN SOURCE,
      '                    CAN BE CONVERTED TO IRRADIANCE- OR APPARENT
      '                    TEMPERATURE VALUES.
      '                    E(N,J,4) = B(J) * V(N,J,4) + A(J)
    return

      '                    READ A SPECIFIED CHANNEL FROM THE A/D CONVERTER
      '                    PARAMETERS USED:
      '                    -- DAC                    A/D CHANNEL TO BE READ
      '                    -- VOLT                   VOLTAGE ON THE CHANNEL

11  out 1813,DAC                                     ' SELECT CHANNEL
    out 1814,0                                       ' START CONVERSION
    wait 1812,128,0                                  ' WAIT FOR CONVERSION READY
    VOLT = INP(1813) + 256*INP(1814)                 ' READ TWO BYTES
    if VOLT > 32767 then
      VOLT = VOLT - 65536!                           ' RANGE  -10 TO +10 VOLT
    end if
    on DAC+1 goto 9,8
    return
9   VOLT = VOLT * 2.100428# - 2040!                  ' CHANNEL 0   X POSITION
    return
8   VOLT = VOLT * 1.994158# - 2040!                  ' CHANNEL 1   Y POSITION
    return

      '                    OUTPUT A VOLTAGE TO A SPECIFIED CHANNEL
      '                    PARAMETERS USED:
      '                    -- DAC = 0           OUTPUT CHANNEL FOR AZIMUTH
      '                    -- DAC = 1           OUTPUT CHANNEL FOR ELEVATION
      '                    -- VX                VOLTAGE ON AZIMUTH CHANNEL
```

```
'                       -- VY               VOLTAGE ON ELEVATION CHANNEL
84 if DAC = 0 then 85
   VY = INT(VY)
   HIGHY = INT(VY/256)                          ' HIGH VY-BYTE
   LOWY = VY - 256*HIGHY
   if HIGHY < 0 then
     HIGHY = HIGHY + 16
   end if
   out 1809,HIGHY                               ' ACTIVATE ELEVATION MOTOR
   out 1808,LOWY
   return

85 VX = INT(VX)
   HIGHX = INT(VX/256)                          ' HIGH VX-BYTE
   LOWX = VX - 256*HIGHX
   if HIGHX < 0 then
     HIGHX = HIGHX + 16
   end if
   out 1811,HIGHX                               ' ACTIVATE AZIMUTH MOTOR
   out 1810,LOWX
   return

'                    TEST IF THE SENSOR HEAD HAS REACHED THE GIVEN
'                    POSITION IF NOT  WAIT FOR IT
'                    CHECK FOR POWER FAILURE WHILE MOVING THE SENSOR HEAD
'                    PARAMETERS USED:
'                    -- VX  AND  VY
'                    -- VMX AND  VMY
86 XYSTOP = 0
   while XSTOP <> 1 or YSTOP <> 1
     gosub 25
     DAC = 0
     if XSTOP <> 1 then
       gosub 11
       VMX = VOLT
       if VMX > VX-5 and VMX < VX+5 then
         XSTOP = 1
       end if
     end if
     DAC = 1
     if YSTOP <> 1 then
       gosub 11
       VMY = VOLT
       if VMY > VY-5 and VMY < VY+5 then
         YSTOP = 1
       end if
     end if
   wend
   XSTOP = 0
   YSTOP = 0
   XYSTOP = 1
   return

'                    START/STOP SCANNING MIRROR AND OPEN HATCH
'                    LOCK/UNLOCK SERVO SYSTEM TO POSITION THE SENSOR HEAD
'                    PARAMETERS USED:
'                    PORTB = 0       START MIRROR, OPEN HATCH AND LOCK
'                    PORTB = 1       STOP MIRROR, CLOSE HATCH AND LOCK
'                    PORTB = 2       START MIRROR, OPEN HATCH AND UNLOCK
'                    PORTB = 3       STOP MIRROR, CLOSE HATCH AND UNLOCK
76 out 1823,145                                 ' WRITE TO CONTROLPORT OF 825
   out 1821,PORTB
   return
```

```
    '                       KEYS 4 AND 5 HANDLE THE PRINTING DURING RUNTIME
49  DIS = 1
    return                                          ' ABLE PRINTING
48  DIS = 0
    return                                          ' DISABLE PRINTING

    '                       E R R O R     H A N D L I N G
    '                       CHECK FOR IRRECOVERABLE ERRORS
40  ERDE = 1
    if ERR = 13 then
      TEXT$ = "  TYPE MISMATCH  "
      goto 38
    elseif ERR = 52 then
      TEXT$ = "  BAD FILE NUMBER  "
      goto 38
    elseif ERR = 53 then
      TEXT$ = "  FILE NOT FOUND  "
      goto 38
    elseif ERR = 57 then
      TEXT$ = "  DEVICE I/O ERROR  "
      goto 38
    elseif ERR = 61 then
      if DRIVE$="A:" then
        TEXT$ = "  DISK A+D FULL  "
        goto 38
      elseif DRIVE$="D:" then
        goto 10
      end if
    elseif ERR = 64 then
      TEXT$ = "  BAD FILE NAME  "
      goto 38
    elseif ERR = 70 then
      TEXT$ = "  DISK WRITE PROTECT  "
      goto 38
    elseif ERR = 71 then
      TEXT$ = "  DISK NOT READY  "
      goto 38
    elseif ERR = 72 then
      TEXT$ = "  DISK MEDIA ERROR  "
      goto 38
    end if

    '                       WRITE RECOVERABLE ERROR MESSAGE TO ERROR FILE
    '                       AND RESUME EXECUTION AT THE NEXT STATEMENT
    if fopen = 0 then
      open DRIVE$+"ERROR.CAR" for append as #3
      fopen = 1
    end if
    write #3,DATE$,TIME$,TEXT$,ERR,ERL
    resume next

    '                       IRRECOVERABLE ERROR DETECTED
    '                       CLOSE THE HATCH AND UNLOCK
38  PORTB = 3
    gosub 76

    '                       DIRECT SENSOR HEAD TO THE FIRST POSITION
    DAC = 0
    VX = X(1)
    gosub 85
    DAC = 1
    VY = Y(1)
    gosub 84
```

```
      '                    START TO ESTABLISH A COMMUNICATION LINK TO A REMOTE
      '                    PC TO PASS ERROR MESSAGES (NOT IMPLEMENTED YET)
    5 if fopen = 0 then
        open DRIVE$+"ERROR.CAR" for append as #3
        fopen = 1
      end if
      write #3,DATE$,TIME$,TEXT$,ERR,ERL

      '                    E N D    O F    P R O G R A M M E
   36 close
      PORTB = 3
      gosub 76
      kill DRIVE$+"DUMMY.CAR"
      if ERDE = 0 then
        kill DRIVE$+"ERROR.CAR"
      end if
   35 cls
      close
      color 7,0
   30 if POWER = 0 then
        locate 10,15 : print "WAITING FOR MAINS POWER TO RETURN."
        AST = 1
        gosub 25
        if POWER = 1 then
          POWER = 0
          goto 26
        end if
        goto 30
      end if
      locate 1,2 : print TEXT$
      EIND$ = DRIVE$ + FIL$
      locate 3,22 : print "       END OF PROGRAMME       "
      color 0,7
      locate 10,22: print "                              "
      locate 11,22: print " Last data has been stored on "
      locate 12,22: print "                              "
      locate 13,22: print "          ";EIND$;"           "
      locate 14,22: print "                              "
      color 7,0
      locate 20,22: print "       HAVE A NICE DAY        "
      end

      '                    CHECK FOR MAINS POWER FAILURE
      '                    TO USE THIS OPTION, THE PC (AND PREFREABLY ALL
      '                    EQUIPMENT) MUST BE BACKED UP WITH A BATTERY SUPPLY,
      '                    TO CONTINUE OPERATION FOR A SHORT WHILE.
   25 MAIN = INP(1820) and 32
      if MAIN = 0 then
        POWER = 1
        if ASS = 0 then
          open "O",#1,"A:POWER.FAL"
            write #1,POWER
          close #1
          ASS = 1
        end if
        AST = 0
        return
      else
        POWER = 0
        ASS = 0
        if PR = 1 and POWER = 0 then
          PR = 0
        end if
      end if
```

```
'                    MAIN = 32 : MAINS POWER FAILURE DETECTED
ERDE = 1
TEXT$ = " MAINS POWER FAILURE "
if AST = 1 then
  AST = 0
  return 30
end if
open "O",#1,"A:POWER.FAL"
  write #1,POWER
close #1
return 38

'                    CHANGE FROM DISK DRIVE D TO A FOR WRITING DATA
10 kill "D:DUMMY.CAR"
close
if ERDE = 0 then
  kill "D:ERROR.CAR"
end if
ERDE = 0
DRIVE$ = "A:"
open "O",#1,"A:RESTUP.CAR"
  write #1,DSTOP$
  write #1,TSTOP$
  write #1,INTV
  write #1,DESC$
  write #1,DRIVE$
close #1
'                    THE FILE ON DRIVE A IS GIVEN THE SAME NAME AS THE
'                    ORIGINAL DATAFILE WITH THE EXTENSION OF  "C" ,
'                    AN ABBREVIATION OF CARABAS
open "A:"+FILENAME$ for append as #2

'                    COPY THE POSITION FILE TO THE NEW DATA DISKETTE
'                          System disk contains position file !
open "O",#1,"A:"+DESC$
print #1,USING "###";NUMBER
print #1,USING "\        \";DAY$
for N = 1 to NUMBER
  print #1 , USING "\                          \" ; POST$(N)
  print #1 , USING "\                        \" ; ELPOST$(N)
  print #1 , USING "+####.##" ; X(N)
  print #1 , USING "+####.##" ; Y(N)
  print #1 , USING "+####.##" ; TETA(N)
  print #1 , USING "+####.##" ; AZIM(N)
  print #1 , USING "+####.##" ; H(N)
  print #1 , USING "+####.##" ; RAN(N)
  print #1 , USING "+####.##" ; EPS(N)
  print #1 , USING "\                          \" ; SCREEN$(N)
  print #1 , USING "\                        \" ; ELSCR$(N)
  print #1 , USING "\                \" ; POSCODE$(N)
next N

'                    OPEN DUMMY FILE WHICH WILL BE KILLED LATER, TO BE ABLE
'                    TO CLOSE FILES IF DISK IS FULL ERROR OCCURS
close #3
open "O",#3,"A:DUMMY.CAR"

  '                  WRITE 2401 BYTES TO THIS FILE
  for JJ = 1 to 100
    write #3,"AAAAAAAAAAAAAAAAAAAA"
  next
close #3

'                    OPEN ERROR COMMUNICATION FILE ON #3
```

```
    open "A:ERROR.CAR" for append as #3
    fopen = 1
    TEL = 0
    resume 61
    end

    '                        MAKE A NEW START TIME AS HH:00:00+N*INTV
 27 ZH = VAL(LEFT$(TIME$,2))
    ZM = VAL(MID$(TIME$,4,2))
    Z = INT(ZM/INTV)
    SECT = 60/INTV
    if (ZH = 23 and ZM >= ((SECT-1)*INTV)-1) or ZH = 24 then
      goto 7
    end if
    for L = 0 to SECT-1
      if L = Z then
        ZM = (L+1)*INTV
      end if
      if ZM = 60 then
        ZM$ = "00"
      elseif ZM < 10 then
        ZM$ = "0" + MID$(STR$(ZM),2,2)
      else
        ZM$ = MID$(STR$(ZM),2,2)
      end if
    next L
    if ZM$ = "00" then
      ZH = ZH + 1
    end if
    if ZH < 10 then
      ZH$ = "0" + MID$(STR$(ZH),2,2)
    else
      ZH$ = MID$(STR$(ZH),2,2)
    end if
    TSTART$ = ZH$ + ":" + ZM$ + ":00"
    DSTART$ = DATE$
    return

    '                        MAKE A NEW START DATE
  7 TSTART$ = "00:00:00"
    DAY = VAL(MID$(DATE$,4,2))
    MONTH = VAL(MID$(DATE$,1,2))
    YEAR = VAL(MID$(DATE$,7,4))
    NDAY = DAY + 1
    on MONTH goto 65,33,65,80,65,80,65,65,80,65,80,65
 33 DAY = 28
    X = YEAR MOD 4
    if X = 0 then
      DAY = 29
    end if
    goto 64
 80 DAY = 30
    goto 64
 65 DAY = 31
 64 if NDAY <= DAY then
      goto 62
    else
      NDAY = NDAY - DAY
      MONTH = MONTH + 1
    end if
    if MONTH > 12 then
      MONTH = 1
      YEAR = YEAR + 1
    end if
```

```
62 if NDAY < 10 then
     DAYST$ = "0" + MID$(STR$(NDAY),2,2)
     goto 59
   end if
   DAYST$ = MID$(STR$(NDAY),2,2)
59 if MONTH < 10 then
     MONTH$ = "0" + MID$(STR$(MONTH),2,2)
     goto 57
   end if
   MONTH$ = MID$(STR$(MONTH),2,2)
57 YEAR$ = MID$(STR$(YEAR),2,4)
   DSTART$ = MONTH$ + "-" + DAYST$ + "-" + YEAR$
   return
```

Appendix D

CARABAS data

```
MAR3090P
 26
04-01-1990
00:00:04
   +5.59
   +5.75
00:00:17
   +9.64
  +10.04
00:00:26
   +8.38
   +9.19
00:00:34
  +10.11
   +9.41
00:00:42
  +14.15
  +14.43
00:00:47
  +22.48
  +22.47
00:00:59
   +5.40
   +4.15
00:01:06
   +5.38
   +4.34
00:01:14
   +5.36
   +4.23
00:01:21
   +5.22
   +4.41
00:01:26
   +5.50
   +4.32
00:01:33
   +7.21
   +4.70
00:01:42
   +7.18
   +6.52
00:01:49
   +8.68
   +6.97
00:01:57
   +5.99
   +6.02
00:02:05
   +6.08
   +5.45
00:02:12
   +5.50
   +5.52
00:02:22
   +7.40
   +6.93
00:02:29
   +9.15
   +9.29
00:02:39
   +9.01
   +9.33
00:02:49
  +11.69
   +9.83
00:02:56
   +7.02
   +2.82
00:03:06
   +7.03
   +3.26
00:03:18
   +8.33
   +8.96
00:03:25
  +10.63
  +10.78
00:03:32
  +11.85
  +11.12
```

Appendix E

Meteorological data

```
04-01-1990,00:05:00, 49
  -24.5,  5
   +0.0,  2
+1029.4, 49
  +88.0, 98
 +283.7, 49
   +0.0, 49
   +2.9, 48
   +0.0,  4
04-01-1990,00:10:00, 52
  -24.5,  1
   +0.0,  1
+1029.4, 52
  +88.7,104
 +283.6, 52
   +0.0, 52
   +3.3, 52
   +0.0,  4
04-01-1990,00:15:00, 52
  -24.5,  3
   +0.0,  4
+1029.4, 52
  +84.6,104
 +283.9, 52
   +0.0, 52
   +2.9, 52
   +0.0,  4
04-01-1990,00:20:00, 52
  -24.5,  4
   +0.0,  1
+1029.4, 52
  +87.4,104
 +283.5, 52
   +0.0, 52
   +2.7, 52
   +0.0,  4
04-01-1990,00:25:00, 52
  -24.5,  1
   +0.0,  1
+1029.4, 52
  +89.6,104
 +283.0, 52
   +0.0, 52
   +2.9, 52
   +0.0,  4
04-01-1990,00:30:00, 52
  -24.5,  6
   +0.0,  1
+1029.5, 52
  +89.1,104
 +282.6, 52
   +0.0, 52
   +3.1, 52
   +0.0,  4
```

UNCLASSIFIED

REPORT DOCUMENTATION PAGE (MOD-NL)

1. DEFENSE REPORT NUMBER (MOD-NL)
TD91-3893

2. RECIPIENT'S ACCESSION NUMBER

3. PERFORMING ORGANIZATION REPORT NUMBER
FEL-91-B201

4. PROJECT/TASK/WORK UNIT NO.
22173

5. CONTRACT NUMBER
-

6. REPORT DATE
NOVEMBER 1991

7. NUMBER OF PAGES
87 (INCLUDING 5 APPENDICES & RDP, EXCLUDING DISTRIBUTION LIST.)

8. NUMBER OF REFERENCES
2

9. TYPE OF REPORT AND DATES COVERED
FINAL REPORT

10. TITLE AND SUBTITLE
CALIBRATED RADIOMETER FOR BACKGROUND SCANNING - CARABAS

11. AUTHOR(S)
ING.R. VAN DER TOUW

12. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
TNO PHYSICS AND ELECTRONICS LABORATORY, P.O. BOX 96864, 2509 JG THE HAGUE
OUDE WAALSDORPERWEG 63, THE HAGUE, THE NETHERLANDS

13. SPONSORING/MONITORING AGENCY NAME(S)
TNO PHYSICS AND ELECTRONICS LABORATORY,
THE HAGUE, THE NETHERLANDS

14. SUPPLEMENTARY NOTES

15. ABSTRACT (MAXIMUM 200 WORDS, 1044 POSITIONS)
THERMAL CHARACTERISTICS OF TARGETS AND BACKGROUNDS DETERMINE IR DETECTABILITY OF THE TARGETS. THEREFORE NOT ONLY KNOWLEDGE OF THE THERMAL BEHAVIOUR OF TARGETS IS NEEDED, BUT ALSO OF BACKGROUND ELEMENTS LIKE TREES AND GRASS. LONG-TERM MEASUREMENTS ARE NECESSARY TO STUDY TIME VARIATIONS OF APPARENT TEMPERATURES, COVERING ALL WEATHER CONDITIONS. FOR THIS REASON A DUAL WAVE-BAND SCANNER IS DEVELOPED FOR AUTONOMOUS RADIATION MEASUREMENTS IN THE SPECTRAL REGIONS 3-5 µM AND 8-12 µM. THIS REPORT DESCRIBES SYSTEM DESIGN AND PERFORMANCE.

16. DESCRIPTORS
THERMAL RADIATION
INFRARED RADIATION MEASUREMENTS
MEASURING INSTRUMENTS

IDENTIFIERS

17a. SECURITY CLASSIFICATION (OF REPORT)
UNCLASSIFIED

17b. SECURITY CLASSIFICATION (OF PAGE)
UNCLASSIFIED

17c. SECURITY CLASSIFICATION (OF ABSTRACT)
UNCLASSIFIED

18. DISTRIBUTION/AVAILABILITY STATEMENT
UNLIMITED

17d. SECURITY CLASSIFICATION (OF TITLES)
UNCLASSIFIED

UNCLASSIFIED